Introduction à la logique

Benjamin Wack (benjamin.wack@univ-grenoble-alpes.fr)

Notes de cours par

Stéphane Devismes Pascal Lafourcade Michel Lévy

Université Grenoble Alpes

Janvier 2023

Organisation

12 semaines

- Cours magistral : 1h30 par semaine
- TD : $2 \times$ 1h30 $=$ 3h par semaine

Cette semaine

- Cours magistral : aujourd'hui ! et jeudi 13h30 comme d'habitude
- TDs : 1 TD seulement cette semaine (horaires exceptionnels)
  2 à partir de la semaine prochaine

https://wackb.gricad-pages.univ-grenoble-alpes.fr/inf402/

Évaluation

Evaluations

- Contrôle continu **60%** : interros **10%**, partiel **20%** et projet **30%**
- Examen : **40%**

Projet par groupes de 3 (ou 4).

- Phase 1 : Modélisation d'un problème en logique (avec interface logicielle)
- Phase 2 : Utilisation d'un solveur SAT pour résoudre ce problème
- Optionnel : Programmation d'un solveur SAT

Exemples de problèmes : N reines, grilles de type Sudoku...

Plan

Préambule

Introduction à la Logique

Logique propositionnelle

Syntaxe

Sens des formules (sémantique)

Conclusion

Logique

Définitions

- La logique précise ce qu'est un raisonnement correct, indépendemment du domaine d'application.
- Un raisonnement est un moyen d'obtenir une conclusion à partir d'hypothèses données.
- Un raisonnement correct ne dit rien sur la vérité des hypothèses, il dit seulement que **de la vérité des hypothèses, on peut déduire la vérité de la conclusion**.

Exemples

Exemple I

- **Hypothèse I :** Tous les hommes sont mortels
- **Hypothèse II :** Socrate est un homme
- **Conclusion :** Socrate est mortel

Exemple II

- **Hypothèse I :** Tout ce qui est rare est cher
- **Hypothèse II :** Un cheval bon marché est rare
- **Conclusion :** Un cheval bon marché est cher !

Ajout d'une hypothèse

Exemple III

- **Hypothèse I :** Tout ce qui est rare est cher
- **Hypothèse II :** Un cheval bon marché est rare
- **Hypothèse III :** Tout ce qui est bon marché n'est pas cher
- **Conclusion :** Hypothèses contradictoires ! Car :
  - **Hypothèse I + hypothèse II :** Un cheval bon marché est cher
  - **Hypothèse III :** Un cheval bon marché n'est pas cher

Applications

- **Hardware** (portes logiques)
- **Vérification et correction des programmes, sécurité** :
  - prouveurs COQ, PVS, Prover9, MACE, . . .
  - applications industrielles (Meteor, Airbus...)
- **Intelligence artificielle** :
  - système expert (*MyCin*), ontologie
- **Programmation** : Prolog
  - intelligence artificielle
  - traitement de la langue
- **Vérification de preuves mathématiques**

Objectifs du cours

- Modéliser et formaliser un problème décrit en langage naturel.
- Comprendre un raisonnement présenté sous forme symbolique, en particulier être capable de déterminer s'il est correct.
- Démontrer, c'est-à-dire construire un raisonnement correct utilisant les règles et/ou les algorithmes de la logique propositionnelle et du premier ordre.
- Écrire une preuve rigoureuse, en particulier par récurrence.

Plan du Semestre

AUJOURD'HUI

- Logique propositionnelle
- Résolution propositionnelle
- Déduction naturelle propositionnelle

PARTIEL

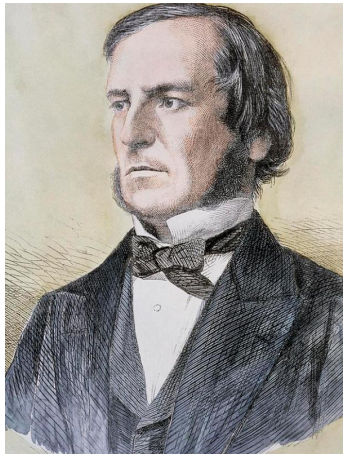
- Logique du premier ordre
- Base de la démonstration automatique (« résolution au premier ordre »)
- Déduction naturelle au premier ordre

EXAMEN

Notion de logique formelle

Idée déjà formulée par Leibniz (voire par les Stoïciens) : pour systématiser le raisonnement, il faut formaliser : les énoncés puis le raisonnement lui-même.

George Boole (1854) suit une approche calculatoire :



- traduire les énoncés par des expressions algébriques
- leur appliquer des règles de calcul qui traduisent le raisonnement humain
- et réinterpréter le résultat

La proposition de Boole donne le véritable départ de la logique mathématique, et en particulier de la logique propositionnelle.

Logique propositionnelle

Définition

La logique propositionnelle est la logique *sans quantificateurs*.
Seules opérations logiques considérées :

- $\neg$ (négation)
- $\wedge$ (conjonction “et”)
- $\vee$ (disjonction “ou”)
- $\Rightarrow$ (implication)
- $\Leftrightarrow$ (équivalence)

Exemple : **Raisonnement formel**

Hypothèses :

- (H1) : Si Pierre est grand, alors Jean n'est pas le fils de Pierre
- (H2) : Si Pierre n'est pas grand, alors Jean est le fils de Pierre
- (H3) : Si Jean est le fils de Pierre alors Marie est la soeur de Jean

Conclusion (C) : Marie est la soeur de Jean ou Pierre est grand.

- p : "Pierre est grand"
- j : "Jean est le fils de Pierre"
- m : "Marie est la soeur de Jean"

- (H1) : $p \Rightarrow \neg j$
- (H2) : $\neg p \Rightarrow j$
- (H3) : $j \Rightarrow m$
- (C) : $m \vee p$

Il s'agira de montrer que $H1 \wedge H2 \wedge H3 \Rightarrow C$:

$$(p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j) \wedge (j \Rightarrow m) \Rightarrow m \vee p$$

est vraie quelque soit la valeur de vérité des propositions p, j, m.

Vocabulaire du langage

- Les constantes : $\top$ (vrai) et $\bot$ (faux)
- Les variables : par exemple x, y_1
- Les connecteurs : $\neg, \vee, \wedge, \Rightarrow, \Leftrightarrow$
- Les parenthèses

Formule (stricte)

Définition 1.1.1

Une formule stricte est définie de manière **inductive** par :

- $\top$ et $\bot$ sont des formules strictes.
- Une variable est une formule stricte.
- Si A est une formule stricte alors $\neg A$ est une formule stricte.
- Si A et B sont des formules strictes et si $\circ$ est un des connecteurs binaires $\vee, \wedge, \Rightarrow, \Leftrightarrow$ alors $(A \circ B)$ est une formule stricte.

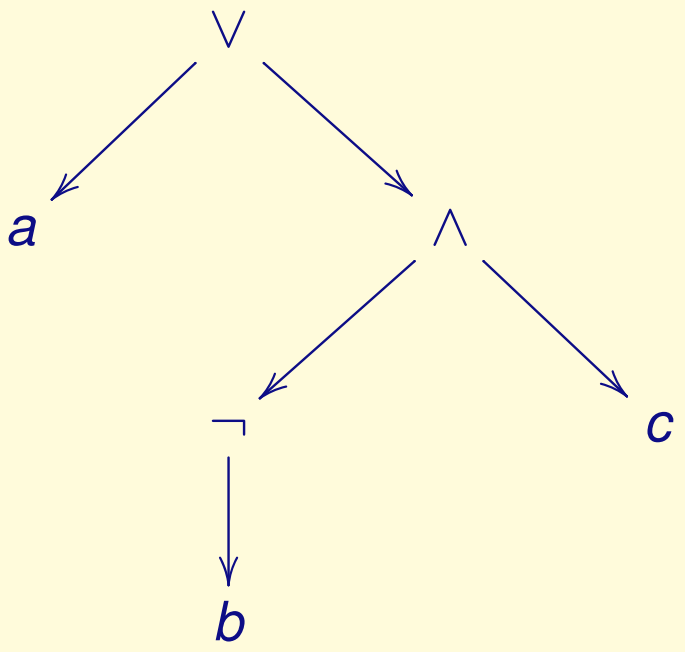
Exemple 1.1.2

$(a \vee (\neg b \wedge c))$ est une formule stricte, mais pas $a \vee (\neg b \wedge c)$, ni $(a \vee (\neg(b) \wedge c))$.

Arbre

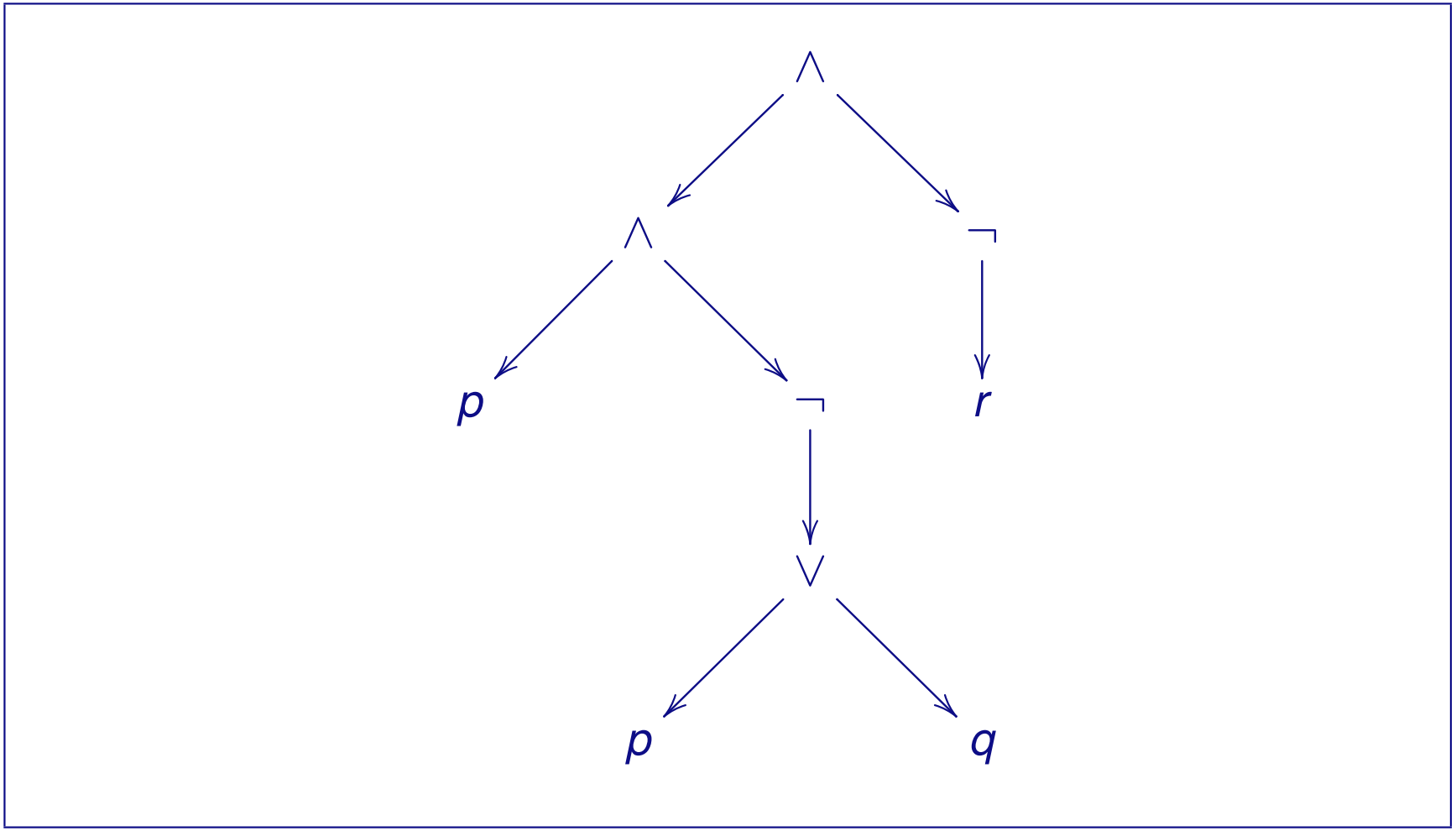
Exemple 1.1.3

La structure de la formule $(a \vee (\neg b \wedge c))$ est mise en évidence par l'arbre suivant :



Exercice

$$((p \wedge \neg(p \vee q)) \wedge \neg r)$$



Taille d'une formule

Définition 1.1.10

La taille d'une formule A, notée $|A|$, est définie inductivement par :

- $|\top| = 0$ et $|\bot| = 0$.
- Si A est une variable alors $|A| = 0$.
- $|\neg A| = 1 + |A|$.
- $|(A \circ B)| = |A| + |B| + 1$.

Exemple 1.1.11

$|(a \vee (\neg b \wedge c))| =$

3.

Premier résultat

Les formules strictes se décomposent de manière unique en leurs sous-formules.

Théorème 1.1.13

Toute formule A s'écrit sous une et une seule de ces formes :

- une variable,
- une constante,
- $\neg B$ où B est une formule,
- $(B \circ C)$ où B et C sont des formules et $\circ$ un connecteur binaire.

Ce qui autorise :

- la démonstration *par cas*
- la récurrence sur la *structure* des formules et non sur leur taille.

Formule à priorité

Définition 1.1.14

Comme pour les formules strictes mais :

- si A et B sont des formules à priorité alors $A \circ B$ est une formule à priorité,
- si A est une formule à priorité alors (A) est une formule à priorité.

En bref : on est libre de placer les parenthèses comme on le souhaite.

Exemple 1.1.15

$a \vee \neg b \wedge c$ est une formule à priorité mais pas une formule stricte.

Règles de priorité

Définition 1.1.16

Par ordre de priorités décroissantes : $\neg$, $\wedge$, $\vee$, $\Rightarrow$ et $\Leftrightarrow$.

Associativité à gauche

Pour deux connecteurs identiques $A \circ B \circ C \ = \ (A \circ B) \circ C$
sauf pour l'implication : $A \Rightarrow B \Rightarrow C \ = \ A \Rightarrow (B \Rightarrow C)$

Exemples de formules à priorité

Exemple 1.1.17

- $a \wedge b \wedge c$ est l'abréviation de

  $((a \wedge b) \wedge c)$

- $a \wedge b \vee c$ est l'abréviation de

  $((a \wedge b) \vee c)$

- $a \vee b \wedge c$ est l'abréviation de

  $(a \vee (b \wedge c))$

Assignation d'une formule

Définition 1.2.1

Une assignation est une fonction qui,
à chaque variable d'une formule, associe une valeur dans $\{0, 1\}$.

$[A]_v$ dénote la valeur de la formule A dans l'assignation v.

Exemple : Soit v une assignation telle que $v(x) = 0$ et $v(y) = 1$.
Appliquer v à $x \vee y$ s'écrit $[x \vee y]_v$.
$[x \vee y]_v = 0 \vee 1 = 1$
Conclusion : $x \vee y$ est vraie pour l'assignation v.

Valeur d'une formule

Définition 1.2.2

Soient A, B des formules, x une variable et v une assignation.

- $[x]_v = v(x)$
- $[\top]_v = 1$, $[\bot]_v = 0$
- $[\neg A]_v = 1 - [A]_v$
- $[(A \vee B)]_v = max\{[A]_v, [B]_v\}$
- $[(A \wedge B)]_v = min\{[A]_v, [B]_v\}$
- $[(A \Rightarrow B)]_v =$ si $[A]_v = 0$ alors 1 sinon $[B]_v$
- $[(A \Leftrightarrow B)]_v =$ si $[A]_v = [B]_v$ alors 1 sinon 0

Table de vérité

Définition 1.2.3

Une table de vérité donne la valeur d'une formule pour **chaque** choix de valeurs des variables de A.

- une ligne de la table de vérité = une assignation
- une colonne = toutes les valeurs d'une formule.

Tables de base

On associe à chaque formule une *valeur* : 0 (faux) ou 1 (vrai).
La constante $\top$ vaut 1 et la constante $\bot$ vaut 0.

Table 1.1 (table de vérité des connecteurs)

x	y	$\neg x$	$x \vee y$	$x \wedge y$	$x \Rightarrow y$	$x \Leftrightarrow y$
0	0	1	0	0	1	1
0	1	1	1	0	1	0
1	0	0	1	0	0	0
1	1	0	1	1	1	1

Exemple :

Exemple 1.2.4

Donner la table de vérité des formules suivantes.

x	y	$x \Rightarrow y$	$\neg x$	$\neg x \vee y$	$(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$	$x \vee \neg y$
0	0	1	1	1	1	1
0	1	1	1	1	1	0
1	0	0	0	0	1	1
1	1	1	0	1	1	1

Formules équivalentes

Définition 1.2.5

Deux formules A et B sont équivalentes (noté $A \equiv B$ ou simplement $A = B$) si elles ont la même valeur pour toute assignation.

Exemple 1.2.6

$x \Rightarrow y \equiv \neg x \vee y$

Remarque :
Le connecteur logique $\Leftrightarrow$ ne signifie pas $A \equiv B$.

Validité, tautologie (1/2)

Définition 1.2.8

- Une formule est valide si elle a la valeur 1 pour toute assignation.
- Aussi appelée une tautologie.
- Noté $\models A$.

Exemple 1.2.9

- $(x \Rightarrow y) \Leftrightarrow (\neg x \vee y)$ est valide ;
- $x \Rightarrow y$ n'est pas valide car

  elle est fausse pour $x = 1$ et $y = 0$.

Validité, tautologie (2/2)

Propriété 1.2.10

Les formules A et B sont équivalentes ($A \equiv B$)
si et seulement si
la formule $A \Leftrightarrow B$ est valide.

Cf table de vérité de $\Leftrightarrow$.

Modèle d'une formule

Définition 1.2.11

Une assignation v qui donne la valeur 1 à une formule est un modèle de cette formule.

On dit aussi que v satisfait A ou v rend A vraie.

Exemple 1.2.12

Un modèle de $x \Rightarrow y$ est :

$x = 1, y = 1$ (il y en a d'autres).

Par contre $x = 1, y = 0$ n'est pas un modèle de $x \Rightarrow y$.

Modèle d'un ensemble de formules

Définition 1.2.13

v est un modèle de l'ensemble $\{A_1, \ldots, A_n\}$
si et seulement si
elle est un modèle de chacune de ces formules.

Exemple 1.2.14

Un modèle de $\{a \Rightarrow b, b \Rightarrow c\}$ est :

$a = 0, b = 0$ (et c quelconque).

Propriété d'un modèle d'un ensemble de formules

Propriété 1.2.15

v est un modèle de $\{A_1, \ldots, A_n\}$
si et seulement si
v est un modèle de $A_1 \wedge \ldots \wedge A_n$.

Exemple 1.2.16

L'ensemble de formules $\{a \Rightarrow b, b \Rightarrow c\}$
et la formule $(a \Rightarrow b) \wedge (b \Rightarrow c)$
ont les mêmes modèles.

Contre-modèle

Définition 1.2.17

Une assignation v qui donne la valeur 0 à A est un contre-modèle de A.

On dit que v ne satisfait pas A ou que v rend la formule fausse.

Exemple 1.2.18

Un contre-modèle de $x \Rightarrow y$ est :

$$\boxed{x = 1, y = 0.}$$

Formule satisfaisable

Définition 1.2.20

Un (ensemble de) formule(s) est satisfaisable s'il admet un modèle.

Définition 1.2.21

Un (ensemble de) formule(s) est insatisfaisable s'il n'est pas satisfaisable.

Exemple 1.2.22

$x \wedge \neg x$ est insatisfaisable, mais $x \Rightarrow y$ est satisfaisable.

Attention

satisfaisable = 1 modèle ou plus	insatisfaisable = 0 modèle
valide = 0 contre-modèle	invalide = 1 contre-modèle ou plus

Aujourd'hui

- Pourquoi définir et utiliser la logique formelle ?
- Formules de logique propositionnelle :
  - 1 variable = 1 proposition (une information) vraie ou fausse
  - 5 connecteurs pour articuler ces propositions
- Sens des formules :
  - assignation = choix d'une valeur de vérité pour chaque variable
  - une formule peut être vraie pour 0, 1, plusieurs ou toutes les assignations

La prochaine fois

Exercice : étudier “Pierre, Jean et Marie” à l’aide des tables de vérité.

- Équivalences remarquables
- Substitutions et remplacements
- Formes normales

Devise d'Oxford

Plus il y a de gruyère, plus il y a de trous
Plus il y a de trous, moins il y a de gruyère
Donc : plus il y a de gruyère, moins il y a de gruyère

Transformation d'une formule logique

Benjamin Wack

Université Grenoble Alpes

Janvier 2023

Au dernier cours

- Pourquoi la logique formelle ?
- Logique propositionnelle
- Syntaxe
- Sens des formules

Notre exemple avec une table de vérité

Hypothèses :

- (H1) : Si Pierre est grand, alors Jean n'est pas le fils de Pierre
- (H2) : Si Pierre n'est pas grand, alors Jean est le fils de Pierre
- (H3) : Si Jean est le fils de Pierre alors Marie est la soeur de Jean

Conclusion (C) : Marie est la soeur de Jean ou Pierre est grand.

$$(p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j) \wedge (j \Rightarrow m) \Rightarrow m \vee p$$

p	j	m	$p \Rightarrow \neg j$	$\neg p \Rightarrow j$	$j \Rightarrow m$	$H_1 \wedge H_2 \wedge H_3$	$m \vee p$	$H_1 \wedge H_2 \wedge H_3 \Rightarrow C$
0	0	0	1	0	1	0	0	1
0	0	1	1	0	1	0	1	1
0	1	0	1	1	0	0	0	1
0	1	1	1	1	1	1	1	1
1	0	0	1	1	1	1	1	1
1	0	1	1	1	1	1	1	1
1	1	0	0	1	0	0	1	1
1	1	1	0	1	1	0	1	1

Plan

Conséquence

Equivalences remarquables

Substitution et remplacement

Formes normales

Conclusion

Conséquence

Définition 1.2.24

A est conséquence de l'ensemble Γ d'hypothèses ($\Gamma \models A$) si tout modèle de Γ est modèle de A.

Remarque 1.2.26

$\models A$ signifie donc bien que A est valide.
(Toute assignation est modèle de l'ensemble vide.)

Exemple de Conséquence

Exemple 1.2.28

$a \Rightarrow b\ ,\ b \Rightarrow c \ \models\ a \Rightarrow c.$

a	b	c	$a \Rightarrow b$	$b \Rightarrow c$	$a \Rightarrow c$
0	0	0	1	1	1
0	0	1	1	1	1
0	1	0	1	0	1
0	1	1	1	1	1
1	0	0	0	1	0
1	0	1	0	1	1
1	1	0	1	0	0
1	1	1	1	1	1

Propriété INCONTOURNABLE

Constamment utilisée dans les exercices et examens.

Propriété 1.2.27

Soit $H_n = A_1 \wedge \ldots \wedge A_n$.
Les trois formulations suivantes sont équivalentes :

1. $A_1, \ldots, A_n \models B$
2. $H_n \Rightarrow B$ est valide.
3. $H_n \wedge \neg B$ est insatisfaisable.

Démonstration.

Elle se base sur les tables de vérité des connecteurs.
On procède en démontrant que $1 \Rightarrow 2$ puis $2 \Rightarrow 3$ et $3 \Rightarrow 1$. $\square$

Preuve (1/3)

- $1 \Rightarrow 2$: supposons que $A_1, \ldots, A_n \models B$.

Soit une assignation v :

- si v n'est pas modèle de $A_1, \ldots, A_n$:
  pour un certain i on a $[A_i]_v = 0$, d'où $[H_n]_v = 0$.
- si v est modèle de $A_1, \ldots, A_n$:
  alors par hypothèse v est un modèle de B donc $[B]_v = 1$.

Dans tous les cas $[H_n \Rightarrow B]_v = 1$, donc $H_n \Rightarrow B$ est valide.

Preuve (2/3)

- $2 \Rightarrow 3$: supposons que $H_n \Rightarrow B$ est valide.
  Pour toute assignation v on a alors :
  - soit $[H_n]_v = 0$,
  - soit $[H_n]_v = 1$ et $[B]_v = 1$.

  Or $[H_n \wedge \neg B]_v = \min([H_n]_v, [\neg B]_v) = \min([H_n]_v, 1 - [B]_v)$.

  Dans les deux cas, nous obtenons $[H_n \wedge \neg B]_v = 0$.
  Donc $H_n \wedge \neg B$ est insatisfaisable.

Preuve (3/3)

- $3 \Rightarrow 1$: supposons que $H_n \wedge \neg B$ est insatisfaisable.
  Montrons que $A_1, \ldots, A_n \models B$. Soit v un modèle de $A_1, \ldots, A_n$:
  - $[H_n]_v = [A_1 \wedge \ldots \wedge A_n]_v = 1$.
  - D'après notre hypothèse $[\neg B]_v = 0$.
    D'où $1 - [B]_v = 0$ et donc $[B]_v = 1$: v est un modèle de B.

La validité du raisonnement par implications circulaires sera démontrée en exercice.

Illustration de la propriété

Exemple 1.2.28

a	b	c	$a \Rightarrow b$	$b \Rightarrow c$	$a \Rightarrow c$	$(a \Rightarrow b) \wedge (b \Rightarrow c) \Rightarrow (a \Rightarrow c)$	$(a \Rightarrow b) \wedge (b \Rightarrow c) \wedge \neg(a \Rightarrow c)$
0	0	0	1	1	1	1	0
0	0	1	1	1	1	1	0
0	1	0	1	0	1	1	0
0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	0
1	0	1	0	1	1	1	0
1	1	0	1	0	0	1	0
1	1	1	1	1	1	1	0

Compacité

Théorème 1.2.30 Compacité propositionnelle

Un ensemble de formules propositionnelles a un modèle si et seulement si tous ses sous-ensembles finis ont un modèle.

Preuve difficile car l'ensemble peut être infini !

Nous l'utiliserons pour la démonstration automatique au premier ordre.

Préambule

Comment prouver qu'une formule est valide ?

- Table de vérité
  - Problème : pour une formule à 100 variables, la table de vérité aura 2^{100} lignes (non calculable même par un ordinateur !).
- Idée :
  - Simplifier la formule en utilisant des règles de calcul
  - Puis étudier la formule simplifiée en utilisant les tables de vérités ou un raisonnement logique

La disjonction

- associative $x \vee (y \vee z) \equiv (x \vee y) \vee z$
- commutative $x \vee y \equiv y \vee x$
- idempotente $x \vee x \equiv x$

Idem pour la conjonction.

Distributivité

- La conjonction est distributive sur la disjonction
  $x \wedge (y \vee z) \equiv (x \wedge y) \vee (x \wedge z)$
- La disjonction est distributive sur la conjonction
  $x \vee (y \wedge z) \equiv (x \vee y) \wedge (x \vee z)$

Neutralité et Absortion

- 0 est l'élément neutre de la disjonction $0 \vee x \equiv x$
- 1 est l'élément neutre de la conjonction $1 \wedge x \equiv x$
- 1 est l'élément absorbant de la disjonction $1 \vee x \equiv 1$
- 0 est l'élément absorbant de la conjonction $0 \wedge x \equiv 0$

Négation

- Les lois de la négation :
  - $x \wedge \neg x \equiv 0$
  - $x \vee \neg x \equiv 1$ (Le tiers-exclus)
- $\neg\neg x \equiv x$
- $\neg 0 \equiv 1$
- $\neg 1 \equiv 0$

Les lois de De Morgan

- $\neg(x \wedge y) \equiv \neg x \vee \neg y$
- $\neg(x \vee y) \equiv \neg x \wedge \neg y$

Augustus De Morgan (1860) étend les notions de l'algèbre de Boole :

- Travail sur les quantificateurs
- Notion de calcul sur des relations
  (voir aussi les travaux de C.S. Peirce)

qui déboucheront sur la logique du premier ordre (cf 2è partie du cours).

- Notion de dualité dans les algèbres de Boole

aujourd'hui connue principalement sous la forme des lois de De Morgan

- Participation (assez anecdotique) aux premières conjectures sur le théorème des quatre couleurs

Lois de simplification

Propriété 1.2.31

Pour tout x, y nous avons :

- $x \vee (x \wedge y) \equiv x$
- $x \wedge (x \vee y) \equiv x$
- $x \vee (\neg x \wedge y) \equiv x \vee y$

Substitution

Définition 1.3.1

Une substitution σ est une fonction de l'ensemble des variables dans l'ensemble des formules.

$A\sigma$ = remplacer dans la formule toute variable x par la formule $\sigma(x)$.

Exemple : $A = \neg(p \wedge q) \Leftrightarrow (\neg p \vee \neg q)$

- Soit σ la substitution suivante : $\sigma(p) = (a \vee b), \sigma(q) = (c \wedge d)$
- $A\sigma = \neg((a \vee b) \wedge (c \wedge d)) \Leftrightarrow (\neg(a \vee b) \vee \neg(c \wedge d))$

Substitution à support fini

Définition 1.3.2

- Le support d'une substitution σ est l'ensemble des variables x telles que $x\sigma \neq x$.
- Une substitution σ à support fini est notée $< x_1 := A_1, \ldots, x_n := A_n >$

Exemple 1.3.3

$A = x \vee x \wedge y \Rightarrow z \wedge y$ et $\sigma =< x := a \vee b, z := b \wedge c >$

$$\boxed{A\sigma = (a \vee b) \vee (a \vee b) \wedge y \Rightarrow (b \wedge c) \wedge y}$$

Propriétés des substitutions

Propriété 1.3.4

Si v est une assignation et σ une substitution :
Soit l'assignation $w : x \mapsto [\sigma(x)]_v$.
Alors pour tout A on a $[A\sigma]_v = [A]_w$.

Exemple 1.3.5 :
Soit $A = x \vee y \vee d$
Soit $\sigma = < x := a \vee b, y := b \wedge c >$
Soit v telle que $v(a) = 1$, $v(b) = 0$, $v(c) = 0$, $v(d) = 0$

$$A\sigma = (a \vee b) \vee (b \wedge c) \vee d$$

$$w(x) = [a \vee b]_v = 1 \vee 0 = 1$$
$$w(y) = [b \wedge c]_v = 0 \wedge 0 = 0$$
$$w(d) = [d]_v = 0$$

$$[A\sigma]_v = (1 \vee 0) \vee (0 \wedge 0) \vee 0$$
$$= 1 \vee 0 \vee 0 = 1$$

$$[A]_w = 1 \vee 0 \vee 0 = 1$$

Cas de base : $|A| = 0$

Deux cas possibles :

- Si A est $\top$ ou $\bot$ alors $A\sigma = A$ et $[A]_v$ ne dépend pas de v.
- Si A est une variable x, par construction $[x\sigma]_v = w(x)$.

Induction

Hypothèse : Supposons que la propriété soit vraie pour toute formule de taille inférieure ou égale à n.
Soit A une formule de taille $n+1$, deux cas possibles :

- Cas 1 : $A = \neg B$ avec $|B| = n$.
  $[A\sigma]_v = [\neg B\sigma]_v = 1 - [B\sigma]_v$ et $[A]_w = [\neg B]_w = 1 - [B]_w$.
  Par hypothèse de récurrence $[B\sigma]_v = [B]_w$
  D'où $[A\sigma]_v = [A]_w$.

Induction

Hypothèse : Supposons que la propriété soit vraie pour toute formule de taille inférieure ou égale à n.
Soit A une formule de taille $n+1$, deux cas possibles :

- Cas 2 : $A = (B \circ C)$ avec $|B| < n+1$ et $|C| < n+1$ et où $\circ$ est un connecteur binaire.
  De même que pour la négation :
  on décompose $[B\sigma \circ C\sigma]_v$ et $[B \circ C]_w$
  et on applique l'hypothèse de récurrence aux formules B et C.

Substitution d'une formule valide

Théorème 1.3.6

Si A est valide alors $A\sigma$ aussi (quelle que soit σ !).

Démonstration.

Soit v une assignation quelconque.

D'après la propriété 1.3.4 : $[A\sigma]_v = [A]_w$ avec $w : x \mapsto [\sigma(x)]_v$.

Puisque A est valide, $[A]_w = 1$.

$A\sigma$ vaut 1 dans toute assignation, c'est donc une formule valide. $\square$

Exemples

Exemple 1.3.7

Soit A la formule $\neg(p \wedge q) \Leftrightarrow (\neg p \vee \neg q)$. Cette formule est valide, c'est une équivalence remarquable. Soit σ la substitution suivante : $< p := (a \vee b), q := (c \wedge d) >$. La formule

$A\sigma = \neg((a \vee b) \wedge (c \wedge d)) \Leftrightarrow (\neg(a \vee b) \vee \neg(c \wedge d))$ est aussi valide.

Remplacement

Définition 1.3.8

La formule D est obtenue en remplaçant dans C certaines occurrences de A par B si :

- C s'écrit $E < x := A >$
- D s'écrit $E < x := B >$

pour une certaine formule E.

Exemples

Exemple 1.3.9

Considérons la formule $C = ((a \Rightarrow b) \vee \neg(a \Rightarrow b))$.

- La formule obtenue en remplaçant toutes les occurrences de $(a \Rightarrow b)$ par $(a \wedge b)$ est

  $$\boxed{D = ((a \wedge b) \vee \neg(a \wedge b))}$$

  avec $E = (x \vee \neg x)$.
- La formule obtenue en remplaçant la *première* occurrence de $(a \Rightarrow b)$ par $(a \wedge b)$ est

  $$\boxed{D = ((a \wedge b) \vee \neg(a \Rightarrow b))}$$

  avec $E = (x \vee \neg(a \Rightarrow b))$.

Propriétés des remplacements

Théorème 1.3.10

Si D est obtenue en remplaçant, dans C, des occurrences de A par B alors : $(A \Leftrightarrow B) \models (C \Leftrightarrow D)$.

Démonstration.

Par définition, $C = E < x := A >$ et $D = E < x := B >$.
Si $[A]_v = [B]_v$ alors les w pour ces deux substitutions sont identiques.
Donc $[C]_v = [D]_v$: l'assignation v est modèle de $(C \Leftrightarrow D)$. $\square$

Exemple 1.3.12 : $p \Leftrightarrow q \models (p \vee (\boxed{p} \Rightarrow r)) \Leftrightarrow (p \vee (\boxed{q} \Rightarrow r))$.

Corollaire 1.3.11

Soit D obtenue en remplaçant, dans C, une occurrence de A par B.
Si $A \equiv B$ alors $C \equiv D$.

Définitions

Définition 1.4.1

- Un littéral est une variable ou la négation d'une variable.
- Un monôme est une conjonction de littéraux.
- Une clause est une disjonction de littéraux.
  (cas particuliers : 0 et 1)

Exemple 1.4.2

- $x, y, \neg z$ sont des littéraux.
- $x \wedge \neg y \wedge z$ est un monôme
- Le monôme $x \wedge \neg y \wedge z \wedge \neg x$ comporte x et $\neg x$: il vaut 0.
- $x \vee \neg y \vee z$ est une clause
- La clause $x \vee \neg y \vee z \vee \neg x$ comporte x et $\neg x$: elle vaut 1.

Forme normale

Definition 1.4.3

Formule en forme normale = seulement $\wedge, \vee$ et des négations sur les variables.

Exemple 1.4.4

La formule $\neg a \vee b$ est en forme normale.
$a \Rightarrow b$ est équivalente mais n'est pas en forme normale.

Toute formule admet une forme normale équivalente.

Mise en forme normale

1. **Élimination des équivalences**
   Remplacer $A \Leftrightarrow B$ par
   (a) $(\neg A \vee B) \wedge (\neg B \vee A)$
   OU
   (b) $(A \wedge B) \vee (\neg A \wedge \neg B)$
2. **Élimination des implications**
   Remplacer $A \Rightarrow B$ par $\neg A \vee B$
3. **Déplacement des négations vers les variables**
   Remplacer
   (a) $\neg\neg A$ par A
   (b) $\neg(A \vee B)$ par $\neg A \wedge \neg B$
   (c) $\neg(A \wedge B)$ par $\neg A \vee \neg B$

Remarque 1.4.5 : simplifications

Simplifier le plus tôt possible :

1. Remplacer $\neg(A \Rightarrow B)$ par $A \wedge \neg B$
2. Remplacer une conjonction par 0 si elle comporte une formule et sa négation
3. Remplacer une disjonction par 1 si elle comporte une formule et sa négation
4. Appliquer :
   - l'idempotence de la conjonction et de la disjonction,
   - le caractère neutre ou absorbant de 0 et de 1,
   - remplacer $\neg 1$ par 0 et $\neg 0$ par 1.
5. Appliquer les simplifications :
   - $x \vee (x \wedge y) = x$,
   - $x \wedge (x \vee y) = x$,
   - $x \vee (\neg x \wedge y) = x \vee y$

Forme normale disjonctive

Définition 1.4.6

Une formule est en forme normale disjonctive (FND) si et seulement si elle est une disjonction (somme) de monômes.

Méthode : distribuer les conjonctions sur les disjonctions
$x \wedge (y \vee z) = (x \wedge y) \vee (x \wedge z)$
L'intérêt des FND est de mettre en évidence les modèles.

Exemple 1.4.7

$(x \wedge y) \vee (\neg x \wedge \neg y \wedge z)$ est une FND, qui a deux modèles principaux :

- $x = 1, y = 1$
- $x = 0, y = 0, z = 1$

Forme normale conjonctive

Définition 1.4.11

Une formule est en forme normale conjonctive (FNC) si et seulement si elle est une conjonction (produit) de clauses.

Appliquer la distributivité (!) de la disjonction sur la conjonction :

- $A \vee (B \wedge C) = (A \vee B) \wedge (A \vee C)$

L'intérêt des FNC est de mettre en évidence les contre-modèles.

Exemple 1.4.12

$(x \vee y) \wedge (\neg x \vee \neg y \vee z)$ est une FNC, qui a deux contre-modèles

- $x = 0, y = 0$
- $x = 1, y = 1, z = 0.$

Utilisée en modélisation (SAT-solvers)

Exemples 1.4.8 et 1.4.13

Mise en FND de :

$$(a \vee b) \wedge (c \vee d \vee e) \equiv$$

$$\boxed{(a \wedge c) \vee (a \wedge d) \vee (a \wedge e) \vee (b \wedge c) \vee (b \wedge d) \vee (b \wedge e).}$$

Mise en FNC de :

$$(a \wedge b) \vee (c \wedge d \wedge e) \equiv$$

$$\boxed{(a \vee c) \wedge (a \vee d) \wedge (a \vee e) \wedge (b \vee c) \wedge (b \vee d) \wedge (b \vee e).}$$

Une autre utilisation de la FND

La mise en forme normale disjonctive permet également de déterminer si une formule est valide ou non.

Soit A une formule dont on souhaite déterminer la validité :

On transforme $\neg A$ en une disjonction de monômes équivalente B :

- Si $B = 0$ alors $\neg A = 0$, donc $A = 1$, c'est-à-dire, A est valide
- Sinon B est égal à une disjonction de monômes non nuls équivalente à $\neg A$, qui nous donnent des modèles de $\neg A$, donc des contre-modèles de A.

Exemple 1.4.9

Soit $A = (p \Rightarrow (q \Rightarrow r)) \Rightarrow (p \wedge q \Rightarrow r)$

Déterminer si A est valide.

$$
\begin{aligned}
&\neg A \\
&\equiv (p \Rightarrow (q \Rightarrow r)) \wedge \neg(p \wedge q \Rightarrow r) \\
&\equiv (\neg p \vee \neg q \vee r) \wedge \neg(p \wedge q \Rightarrow r) \\
&\equiv (\neg p \vee \neg q \vee r) \wedge (p \wedge q \wedge \neg r) \\
&\equiv (\neg q \vee r) \wedge p \wedge q \wedge \neg r \\
&\equiv (r) \wedge p \wedge q \wedge \neg r \\
&= 0
\end{aligned}
$$

Donc $\neg A = 0$ et $A = 1$, c'est-à-dire A est valide.

Exemple 1.4.10

$$
\begin{aligned}
&\neg A \\
&= \neg((a \Rightarrow b) \wedge c) \wedge \neg(a \wedge d) && \text{(de Morgan)} \\
&= (\neg(a \Rightarrow b) \vee \neg c) \wedge (\neg a \vee \neg d) && \text{(de Morgan)} \\
&= ((a \wedge \neg b) \vee \neg c) \wedge (\neg a \vee \neg d) && (\neg(\ldots \Rightarrow \ldots)) \\
&= (a \wedge \neg b \wedge \neg a) \vee (a \wedge \neg b \wedge \neg d) \vee (\neg c \wedge \neg a) \vee (\neg c \wedge \neg d) \\
& && \text{(distributivité } \vee \text{ sur } \wedge) \\
&= (a \wedge \neg b \wedge \neg d) \vee (\neg c \wedge \neg a) \vee (\neg c \wedge \neg d) && \text{(1er monôme contradictoire)}
\end{aligned}
$$

On obtient 3 modèles de $\neg A$: $(a = 1, b = 0, d = 0)$, $(a = 0, c = 0)$, $(c = 0, d = 0)$.

C'est-à-dire, des contre-modèles de A.

Donc A n'est pas valide.

Aujourd'hui

- Les substitutions permettent de **déduire la validité** d'une formule à partir d'une autre
- Les remplacements permettent de modifier une partie d'une formule **sans changer sa signification** et autorisent donc à effectuer des calculs sur les formules
- Toute formule admet des formes normales qui permettent d'**identifier ses modèles** ou ses contre-modèles

La prochaine fois

- Algèbre de Boole
- Fonctions booléennes
- Résolution

À chercher : prouver par simplification de formule notre exemple

$$(p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j) \wedge (j \Rightarrow m) \Rightarrow m \vee p$$

Résolution Propositionnelle

Système de déduction

Benjamin Wack

Stéphane Devismes Pascal Lafourcade Michel Lévy

Université Grenoble Alpes

Janvier 2023

Au dernier cours

- Équivalences remarquables
- Substitutions et remplacements
- Formes normales

Jean, Pierre et Marie par simplification

$$(p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j) \wedge (j \Rightarrow m) \Rightarrow m \vee p$$

$$\neg((p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j) \wedge (j \Rightarrow m)) \vee m \vee p$$

$$\neg(p \Rightarrow \neg j) \vee \neg(\neg p \Rightarrow j) \vee \neg(j \Rightarrow m) \vee m \vee p$$

$$(p \wedge \neg\neg j) \vee (\neg p \wedge \neg j) \vee (j \wedge \neg m) \vee m \vee p$$

avec $x \vee (x \wedge y) \equiv x$

$$(\neg p \wedge \neg j) \vee (j \wedge \neg m) \vee m \vee p$$

avec $x \vee (\neg x \wedge y) \equiv x \vee y$

$$\neg j \vee j \vee m \vee p = 1$$

Exemples 1.4.8 et 1.4.13

Mise en FND de :

$$(a \vee b) \wedge (c \vee d \vee e) \equiv$$

$$\boxed{(a \wedge c) \vee (a \wedge d) \vee (a \wedge e) \vee (b \wedge c) \vee (b \wedge d) \vee (b \wedge e).}$$

Mise en FNC de :

$$(a \wedge b) \vee (c \wedge d \wedge e) \equiv$$

$$\boxed{(a \vee c) \wedge (a \vee d) \wedge (a \vee e) \wedge (b \vee c) \wedge (b \vee d) \wedge (b \vee e).}$$

Une autre utilisation de la FND

La mise en forme normale disjonctive permet également de déterminer si une formule est valide ou non.

Soit A une formule dont on souhaite déterminer la validité :

On transforme $\neg A$ en une disjonction de monômes équivalente B :

- Si $B = 0$ alors $\neg A = 0$, donc $A = 1$, c'est-à-dire, A est valide
- Sinon B est égal à une disjonction de monômes non nuls équivalente à $\neg A$, qui nous donnent des modèles de $\neg A$, donc des contre-modèles de A.

Exemple 1.4.9

Soit $A = (p \Rightarrow (q \Rightarrow r)) \Rightarrow (p \wedge q \Rightarrow r)$

Déterminer si A est valide.

$$
\begin{aligned}
&\neg A \\
&\equiv (p \Rightarrow (q \Rightarrow r)) \wedge \neg(p \wedge q \Rightarrow r) \\
&\equiv (\neg p \vee \neg q \vee r) \wedge \neg(p \wedge q \Rightarrow r) \\
&\equiv (\neg p \vee \neg q \vee r) \wedge (p \wedge q \wedge \neg r) \\
&\equiv (\neg q \vee r) \wedge p \wedge q \wedge \neg r \\
&\equiv (r) \wedge p \wedge q \wedge \neg r \\
&= 0
\end{aligned}
$$

Donc $\neg A = 0$ et $A = 1$, c'est-à-dire A est valide.

Exemple 1.4.10

Soit $A = (a \Rightarrow b) \wedge c \vee (a \wedge d)$.

Déterminer si A est valide.

$$
\begin{array}{ll}
\neg A & \\
= \neg((a \Rightarrow b) \wedge c) \wedge \neg(a \wedge d) & \text{(de Morgan)} \\
= (\neg(a \Rightarrow b) \vee \neg c) \wedge (\neg a \vee \neg d) & \text{(de Morgan)} \\
= ((a \wedge \neg b) \vee \neg c) \wedge (\neg a \vee \neg d) & (\neg(\ldots \Rightarrow \ldots)) \\
= (a \wedge \neg b \wedge \neg a) \vee (a \wedge \neg b \wedge \neg d) \vee (\neg c \wedge \neg a) \vee (\neg c \wedge \neg d) & \\
 & \text{(distributivité } \vee \text{ sur } \wedge) \\
= (a \wedge \neg b \wedge \neg d) \vee (\neg c \wedge \neg a) \vee (\neg c \wedge \neg d) & \text{(1er monôme contradictoire)}
\end{array}
$$

On obtient 3 modèles de $\neg A$: $(a = 1, b = 0, d = 0)$, $(a = 0, c = 0)$, $(c = 0, d = 0)$.
C'est-à-dire, des contre-modèles de A.
Donc A n'est pas valide.

Plan

Algèbre de Boole

Fonctions booléennes

L'outil BDDC

Introduction à la résolution

Quelques définitions et notations

Conclusion

Définition 1.5.1

Une algèbre de Boole est un ensemble :

- d'au moins deux éléments (0 et 1)
- avec trois opérations : complément ($\overline{x}$), somme ($+$) et produit (.)
- telles que :

1. la somme est associative, commutative, élément neutre 0
2. le produit est associatif, commutatif, élément neutre 1
3. le produit est distributif sur la somme
4. la somme est distributive sur le produit
5. les lois du complément :
   - $x + \overline{x} = 1$,
   - $x.\overline{x} = 0$.

La logique propositionnelle est une algèbre de Boole

Les axiomes se démontrent par tables de vérités.

Autre exemple :

Algèbre de Boole	$\mathcal{P}(X)$
1	X
0	$\emptyset$
$\overline{p}$	$X - p$
$p + q$	$p \cup q$
$p.q$	$p \cap q$

Propriétés d'une algèbre de Boole

Propriété 1.5.3

- Pour tout x, il y a un et un seul y tel que $x + y = 1$ et $xy = 0$, autrement dit le complément est unique.
-
  1. $\overline{1} = 0$
  2. $\overline{0} = 1$
  3. $\overline{\overline{x}} = x$
  4. $x.x = x$
  5. $x + x = x$
  6. $1 + x = 1$
  7. $0.x = 0$
  8. Lois de De Morgan :
     - $\overline{xy} = \bar{x} + \bar{y}$
     - $\overline{x + y} = \bar{x}.\bar{y}$

Preuve

1. $\overline{1} = 0$.

   Comme $x.\overline{x} = 0$, on a $1.\overline{1} = 0$.
   Puisque 1 est neutre, $\overline{1} = 0$.

2. $\overline{0} = 1$.

   Idem : $x + \overline{x} = 1$ donc $0 + \overline{0} = 1$.
   Puisque 0 est neutre, $\overline{0} = 1$.

3. $\overline{\overline{x}} = x$.

   Par commutativité, $\overline{x} + x = 1$ et $\overline{x}.x = 0$
   Par unicité de la négation de $\overline{x}$, on a $\overline{\overline{x}} = x$.

Preuve

4. Idempotence du produit : $x.x = x$.

$$\begin{aligned} x &= x.1 \\ &= x.(x+\overline{x}) \\ &= x.x + x.\overline{x} \\ &= x.x + 0 \\ &= x.x \end{aligned}$$

5. Idempotence de la somme : $x + x = x$

Même principe en partant de $x = x + 0$

Preuve

6. 1 est élément absorbant de la somme : $1 + x = 1$.

Par idempotence de la somme :

$$\begin{aligned} 1 + x &= (x + \overline{x}) + x \\ &= x + \overline{x} \\ &= 1 \end{aligned}$$

7. 0 est élément absorbant du produit : $0.x = 0$.

De même en partant de $0.x = (x.\overline{x}).x$

8. Lois de De Morgan : $\overline{xy} = \bar{x} + \bar{y}$

Nous montrons que $xy + (\bar{x} + \bar{y}) = 1$

$$\begin{aligned} x.y + (\overline{x} + \overline{y}) &= (x + \overline{x} + \overline{y}).(y + \overline{x} + \overline{y}) \\ &= (1 + \overline{y}).(1 + \overline{x}) \\ &= 1.1 \\ &= 1 \end{aligned}$$

De même $xy.(\overline{x} + \overline{y}) = 0$.
Par unicité de la négation $\overline{x} + \overline{y}$ est la négation de $x.y$.

De même on montre que $\overline{x + y} = \overline{x}.\overline{y}$ en inversant les rôles joués par . et $+$

Définition 1.6.1 : Fonction booléenne

Une fonction booléenne est une fonction dont les arguments et le résultat sont dans le domaine $\mathbb{B} = \{0, 1\}$.

Exemple 1.6.2

Lesquelles de ces fonctions sont booléennes ?

- $f : \mathbb{B} \to \mathbb{B} : f(x) = \neg x$
- $f : \mathbb{N} \to \mathbb{B} : f(x) = x \bmod 2$
- $f : \mathbb{B} \to \mathbb{N} : f(x) = x + 1$
- $f : \mathbb{B} \times \mathbb{B} \to \mathbb{B} : f(x, y) = \neg(x \wedge y)$

Fonctions booléennes et somme de monômes

Théorème 1.6.3

On pose $x^0 = \bar{x}$ et $x^1 = x$.

Pour une fonction booléenne f à n arguments on pose :

$$A = \sum_{f(a_1,\ldots,a_n)=1} x_1^{a_1} \ldots x_n^{a_n}.$$

A est la somme des monômes $x_1^{a_1} \ldots x_n^{a_n}$ tels que $f(a_1, \ldots, a_n) = 1$.

Pour toute assignation v telle que $v(x_1) = a_1, \ldots, v(x_n) = a_n$,
on a $f(a_1, \ldots, a_n) = [A]_v$.

Exemple 1.6.4

La fonction *maj* à 3 arguments vaut 1 lorsqu'au moins 2 de ses arguments valent 1.

Définir la somme de monômes équivalente (théorème 1.6.3)

x_1	x_2	x_3	$maj(x_1, x_2, x_3)$	
0	0	0	0	
0	0	1	0	
0	1	0	0	
0	1	1	1	$\overline{x_1}x_2x_3$
1	0	0	0	
1	0	1	1	$x_1\overline{x_2}x_3$
1	1	0	1	$x_1x_2\overline{x_3}$
1	1	1	1	$x_1x_2x_3$

$$\boxed{maj(x_1, x_2, x_3) = \overline{x_1}x_2x_3 + x_1\overline{x_2}x_3 + x_1x_2\overline{x_3} + x_1x_2x_3}$$

Vérifions le théorème 1.6.3 sur l'exemple 1.6.4

x_1	x_2	x_3	$maj(x_1, x_2, x_3)$	$\overline{x_1}\,x_2\,x_3$	$x_1\,\overline{x_2}\,x_3$	$x_1\,x_2\,\overline{x_3}$	$x_1\,x_2\,x_3$	$\overline{x_1}\,x_2\,x_3 + x_1\,\overline{x_2}\,x_3 + x_1\,x_2\,\overline{x_3} + x_1\,x_2\,x_3$
0	0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0
0	1	1	1	1	0	0	0	1
1	0	0	0	0	0	0	0	0
1	0	1	1	0	1	0	0	1
1	1	0	1	0	0	1	0	1
1	1	1	1	0	0	0	1	1

$$\boxed{maj(x_1, x_2, x_3) = \overline{x_1}\,x_2\,x_3 + x_1\,\overline{x_2}\,x_3 + x_1\,x_2\,\overline{x_3} + x_1\,x_2\,x_3}$$

Fonctions booléennes et produit de clauses

Théorème 1.6.5

Pour une fonction booléenne f à n arguments on pose :

$$A = \prod_{f(a_1,\ldots,a_n)=0} x_1^{\overline{a_1}} + \ldots + x_n^{\overline{a_n}}.$$

A est le produit des clauses $x_1^{\overline{a_1}} + \ldots + x_n^{\overline{a_n}}$ telles que $f(a_1,\ldots,a_n) = 0$.

Pour toute assignation v telle que $v(x_1) = a_1, \ldots, v(x_n) = a_n$,
on a $f(a_1,\ldots,a_n) = [A]_v$.

Exemple 1.6.6

La fonction *maj* à 3 arguments vaut 1 lorsqu'au moins 2 de ses arguments valent 1.

Définir le produit de clauses équivalent (théorème 1.6.5)

x_1	x_2	x_3	$maj(x_1, x_2, x_3)$
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

$x_1 + x_2 + x_3$

$x_1 + x_2 + \overline{x_3}$

$x_1 + \overline{x_2} + x_3$

$\overline{x_1} + x_2 + x_3$

$$maj(x_1, x_2, x_3) = (x_1 + x_2 + x_3)(x_1 + x_2 + \overline{x_3})(x_1 + \overline{x_2} + x_3)(\overline{x_1} + x_2 + x_3)$$

Vérifions le théorème 1.6.5 sur l'exemple 1.6.6

x_1	x_2	x_3	$maj(x_1, x_2, x_3)$	$x_1 + x_2 + x_3$	$x_1 + x_2 + \overline{x_3}$	$x_1 + \overline{x_2} + x_3$	$\overline{x_1} + x_2 + x_3$	$(x_1 + x_2 + x_3)$ $(x_1 + x_2 + \overline{x_3})$ $(x_1 + \overline{x_2} + x_3)$ $(\overline{x_1} + x_2 + x_3)$
0	0	0	0	0	1	1	1	0
0	0	1	0	1	0	1	1	0
0	1	0	0	1	1	0	1	0
0	1	1	1	1	1	1	1	1
1	0	0	0	1	1	1	0	0
1	0	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1
1	1	1	1	1	1	1	1	1

$$\boxed{maj(x_1, x_2, x_3) = (x_1 + x_2 + x_3)(x_1 + x_2 + \overline{x_3})(x_1 + \overline{x_2} + x_3)(\overline{x_1} + x_2 + x_3)}$$

BDDC (*Binary Decision Diagram based Calculator*)

BDDC est un outil pour la manipulation de formules propositionnelles développé par Pascal Raymond et disponible à l'adresse suivante :
http://www-verimag.imag.fr/~raymond/home/tools/bddc/

Plan du Semestre

- Logique propositionnelle *
- Résolution propositionnelle
- Déduction naturelle propositionnelle

PARTIEL

- Logique du premier ordre
- Base de la démonstration automatique (« résolution au premier ordre »)
- Déduction naturelle au premier ordre

EXAMEN

Méthodes de déduction

- Une formule est-elle valide ?
- Un raisonnement est-il correct ?

Deux méthodes :

Les tables de vérités et les transformations

Problème

Si le nombre de variables augmente, ces méthodes sont très longues

Exemple

Par une table de vérité, pour vérifier
$a \Rightarrow b, b \Rightarrow c, c \Rightarrow d, d \Rightarrow e, e \Rightarrow f, f \Rightarrow g, g \Rightarrow h, h \Rightarrow i, i \Rightarrow j \models a \Rightarrow j$
il faut tester $2^{10} = 1024$ lignes.

Or, par déduction, ce raisonnement est correct :

1. Par transitivité de l'implication, il suffit de montrer que $a \Rightarrow j \models a \Rightarrow j$.
2. Or par définition, la formule $a \Rightarrow j$ est une conséquence d'elle-même.

- Formalisation d'un système de déduction

David Hilbert (1862-1943)

- Fondateur de l'école **formaliste** : les mathématiques peuvent et doivent être formalisées pour être démontrées indiscutablement.
- Programme de Hilbert (1920) :
  « *Wir müssen wissen. Wir werden wissen.* »
  en réponse à « *Ignoramus et ignorabimus* »

  - choisir un ensemble fini d'axiomes suffisant pour exprimer toutes les mathématiques
  - démontrer la cohérence de cet ensemble
  - élaborer un algorithme qui décide si une proposition est démontrable (*Entscheidungsproblem*)
- Système *à la Hilbert* : des axiomes comme $\vdash p \Rightarrow (q \Rightarrow p)$
  et quelques règles de déduction comme $\dfrac{\vdash p \Rightarrow q \qquad \vdash p}{\vdash q}$
- Démonstrations complètes mais fastidieuses à écrire et à relire

Aujourd'hui : résolution propositionnelle

- 1 seule règle : la résolution

$$a + \overline{b}, b + c \models a + c$$

  pour des formules mises en FNC (conjonction de clauses)
- Notion formelle de preuve par résolution
- Quelques propriétés de la résolution

Définitions

Définition 2.1.1

On assimile une clause à l'ensemble de ses littéraux (sans doublon, non ordonné). Ainsi on dira que :

- Un littéral est élément d'une clause.
- Une clause A est incluse dans une clause B (ou sous-clause)
- Deux clauses sont égales si elles ont les mêmes littéraux.

Exemple 2.1.2

- Les clauses $p+\overline{q}$, $\overline{q}+p$, et $p+\overline{q}+p$ sont égales
- $p \in \overline{q}+p+r+p$
- $p+\overline{q} \subseteq \overline{q}+p+r+p$
- $\overline{q}+p+r+p - p = \overline{q}+r$
- $p+p+p - p = \bot$
- Ajouter le littéral r à la clause p donne la clause $p+r$
- Ajouter le littéral p à la clause $\bot$ donne la clause p

Littéral complémentaire

Définition 2.1.4

Nous notons L^c le littéral complémentaire d'un littéral L :

Si L est une variable, L^c est sa négation.

Si L est la négation d'une variable, L^c est la même sans négation.

Exemple 2.1.5

$x^c = \overline{x}$ et $\overline{x}^c = x$.

Résolvant

Définition 2.1.6

Soient A et B deux clauses.

La clause C est un résolvant de A et B ssi il existe un littéral L tel que

$$L \in A, \qquad L^c \in B, \qquad C = (A - \{L\}) \cup (B - \{L^c\})$$

"C est un résolvant de A et B" est représenté par :

$$\frac{A \qquad B}{C}$$

C est engendrée par A et B.
A et B sont les parents de la clause C.

Exemples de résolution

Exemple 2.1.7

Donnez les résolvants de :

- $p+q+r$ et $p+\bar{q}+r$

$$\frac{p+q+r \qquad p+\bar{q}+r}{p+r}$$

- $p+\bar{q}$ et $\bar{p}+q+r$

$$\frac{p+\bar{q} \qquad \bar{p}+q+r}{\bar{p}+p+r} \qquad \frac{p+\bar{q} \qquad \bar{p}+q+r}{\bar{q}+q+r}$$

- p et $\bar{p}$

$$\frac{p \qquad \bar{p}}{\perp}$$

Propriété

Propriété 2.1.8

Si l'un des parents d'un résolvant est valide, le résolvant est valide ou contient l'autre parent.

Preuve.

Cette preuve est demandée dans l'exercice 39. □

Exemple

$$\frac{p+\bar{p}+q \qquad \bar{q}+r}{p+\bar{p}+r} \qquad \frac{p+\bar{p}+q \qquad \bar{p}+r}{\bar{p}+q+r}$$

Notion de preuve

Définition 2.1.11

Soient Γ un ensemble de clauses et C une clause.

Une preuve de C à partir de Γ est une liste de clauses où :

- toute clause est un élément de Γ
- ou est un résolvant de deux clauses la précédant
- la dernière clause est C.

La clause C est déduite de Γ, notée $\Gamma \vdash C$, s'il y a une preuve de C à partir de Γ.

La taille d'une preuve est le nombre de lignes qu'elle contient.

Exemple de preuve

Exemple 2.1.12

Soit Γ l'ensemble de clauses $\overline{p}+q, p+\overline{q}, \overline{p}+\overline{q}, p+q$.

Nous montrons que $\Gamma \vdash \perp$:

1	$p+q$	Hypothèse
2	$p+\bar{q}$	Hypothèse
3	p	Résolvant de 1, 2
4	$\bar{p}+q$	Hypothèse
5	q	Résolvant de 3, 4
6	$\bar{p}+\bar{q}$	Hypothèse
7	$\bar{p}$	Résolvant de 5, 6
8	$\perp$	Résolvant de 3, 7

Preuve en arbre

Exemple 2.1.12

Soit Γ l'ensemble de clauses $\bar{p}+q, p+\bar{q}, \bar{p}+\bar{q}, p+q$.

Nous montrons que $\Gamma \vdash \bot$:

$$\dfrac{\dfrac{\dfrac{\dfrac{p+q \qquad p+\bar{q}}{p} \qquad \bar{p}+q}{q} \qquad \bar{p}+\bar{q}}{\bar{p}} \qquad \dfrac{p+q \qquad p+\bar{q}}{p}}{\bot}$$

Monotonie et Composition

Propriété 2.1.14

1. Monotonie : Si $\Gamma \vdash A$ et si $\Gamma \subseteq \Delta$ alors $\Delta \vdash A$
2. Composition : Si $\Gamma \vdash A$ et $\Gamma \vdash B$ et si C est un résolvant de A et B alors $\Gamma \vdash C$.

Preuve.

Exercice 38 □

Aujourd'hui

- Les **équivalences remarquables** correspondent à des règles de calcul dans une algèbre de Boole
- Toute fonction booléenne peut être représentée par une formule (en forme normale)

- Un système de déduction est composé d'un ensemble de **règles formelles**
- Une preuve est une suite d'applications de ces règles à partir d'**hypothèses**

La prochaine fois

- Cohérence et Complétude du système
- Stratégie complète
- Davis et Putnam

À préparer

Hypothèses :

- (H1) : Si Pierre est grand, alors Jean n'est pas le fils de Pierre
- (H2) : Si Pierre n'est pas grand, alors Jean est le fils de Pierre
- (H3) : Si Jean est le fils de Pierre alors Marie est la soeur de Jean

Conclusion (C) : Marie est la soeur de Jean ou Pierre est grand.

Transformer en clauses les hypothèses et $\neg C$.

Que peut-on (doit-on) **démontrer** par résolution ?

Résolution Propositionnelle

Deuxième Partie : Algorithmes

Benjamin Wack

Université Grenoble Alpes

Janvier 2023

Preuve par résolution de l'exemple du cours

- (H1) : $p \Rightarrow \neg j \;\equiv\; \neg p \vee \neg j$
- (H2) : $\neg p \Rightarrow j \;\equiv\; p \vee j$
- (H3) : $j \Rightarrow m \;\equiv\; \neg j \vee m$
- ($\neg$ C) : $\neg m \wedge \neg p$

Clauses : $\{\neg p \vee \neg j,\ p \vee j,\ \neg j \vee m,\ \neg m,\ \neg p\}$

$$\dfrac{\dfrac{\dfrac{p \vee j \qquad \neg j \vee m}{p \vee m} \qquad \neg m}{p} \qquad \neg p}{\bot} \quad \text{OU} \quad \dfrac{\dfrac{\dfrac{p \vee j \qquad \neg p}{j} \qquad \neg j \vee m}{m} \qquad \neg m}{\bot}$$

OU...

Au dernier cours

- Algèbre de Boole
- Fonctions booléennes
- Résolution

(1) $A \vdash B$

B se *déduit* de A : il existe une preuve par résolution de B à partir de A.

(2) $A \models B$

B est *conséquence* de A : tout modèle de A est aussi un modèle de B.

Aujourd'hui : Cohérence

(1) $\Rightarrow$ (2)

Aujourd'hui : Complétude

(2) $\Rightarrow$ (1)

Plan

Cohérence

Complétude

Introduction aux algorithmes de résolution

Davis, Putnam, Logemann et Loveland

Stratégie complète

Conclusion

Définition

La cohérence d'un système logique, c'est le fait que les preuves obtenues dans ce système « ne prouvent que des choses vraies ».

Cohérence de la règle de résolution

Théorème 2.1.15

Si C est un résolvant de A et B alors $A, B \models C$.

Preuve.

Si C est un résolvant de A et B, alors il y a un littéral L tel que
$L \in A, L^c \in B$, et $C = (A - \{L\}) \cup (B - \{L^c\})$.

Soit v telle que $[A]_v = 1$ et $[B]_v = 1$: montrons que $[C]_v = 1$.

- Si $[L]_v = 1$. Alors $[L^c]_v = 0$.
  Comme $[B]_v = 1$, v est modèle d'un littéral de $(B - \{L^c\})$. Donc $[C]_v = 1$.
- Si $[L^c]_v = 1$. Alors $[L]_v = 0$.
  Comme $[A]_v = 1$, v est modèle de $(A - \{L\})$. Donc $[C]_v = 1$.

Dans tous les cas v est modèle de C.

□

Cohérence de la déduction

Théorème 2.1.16

Soit Γ un ensemble de clauses et C une clause. Si $\Gamma \vdash C$ alors $\Gamma \models C$.

Preuve.

Soit P une preuve de C à partir de Γ.
Supposons que pour toute preuve $\Gamma \vdash D$ plus courte que P, nous avons $\Gamma \models D$.
Montrons que $\Gamma \models C$. Nous avons deux cas possibles :

1. C est un élément de Γ, dans ce cas évidemment $\Gamma \models C$.
2. $\Gamma \vdash A$ et $\Gamma \vdash B$ (avec une preuve plus courte) et

$$\frac{A \qquad B}{C}$$

   Par hypothèse de récurrence : $\Gamma \models A$ et $\Gamma \models B$.
   Par cohérence de la règle de résolution : $A, B \models C$. Donc $\Gamma \models C$.

$\square$

Définition

Complétude pour la réfutation : Si $\Gamma \models \bot$ alors $\Gamma \vdash \bot$.

Nous démontrons ce résultat pour Γ fini.

$\Gamma[L := 1]$

Définition 2.1.18

Soient Γ un ensemble de clauses et L un littéral.

$\Gamma[L := 1]$ est obtenu en :

- supprimant les clauses contenant L
- enlevant L^c des autres clauses

De même $\Gamma[L := 0]$ en échangeant les rôles de L et L^c.

Remarque : on a une variable de moins.

Exemples

Exemple 2.1.19

$\Gamma = \overline{p}+q, \overline{q}+r, p+q, p+r.$

- $\Gamma[p := 1] =$

  $\{q, \overline{q}+r\}.$

- $\Gamma[p := 0] =$

  $\{\overline{q}+r, q, r\}.$

Observons que :

- $(\overline{1}+q)(\overline{q}+r)(1+q)(1+r) =$

  $q(\overline{q}+r)$

- $(\overline{0}+q)(\overline{q}+r)(0+q)(0+r) =$

  $(\overline{q}+r)qr$

Propriété de $\Gamma[L := \ldots]$

Propriété 2.1.21

Γ a un modèle si et seulement si $\Gamma[L := 1]$ ou $\Gamma[L := 0]$ en a un.

Preuve.

$\Rightarrow$ Si v est un modèle de Γ alors c'est un modèle de l'un des $\Gamma[L := \ldots]$ (à choisir selon $[L]_v$).

$\Leftarrow$ Si v est un modèle de $\Gamma[L := i]$ alors on peut en déduire un modèle de Γ (en posant $[L]_{v'} = i$).

$\square$

Lemme 2.1.22

Lemme 2.1.22

Soit Γ un ensemble de clauses, C une clause et L un littéral.
Si $\Gamma[L := 1] \vdash C$ alors $\Gamma \vdash C$ ou $\Gamma \vdash C + L^c$.

Preuve.

On rajoute le littéral L^c aux clauses où on l'avait enlevé dans Γ.

- Si $C \in \Gamma[L := 1]$:
  - soit C était dans Γ alors $\Gamma \vdash C$
  - soit C est obtenue en enlevant un L^c alors $\Gamma \vdash C + L^c$
- Si C est un résolvant de A et B :
  - soit $\Gamma \vdash A$ et $\Gamma \vdash B$ d'où $\Gamma \vdash C$
  - soit il faut rajouter L^c dans A ou B, donc dans C aussi

$\square$

Complétude de la résolution

Théorème 2.1.24

Soit Γ un ensemble fini de clauses. Si $\Gamma \vDash \bot$ alors $\Gamma \vdash \bot$.

Preuve

Par récurrence sur le nombre de variables de Γ.

- Cas de base : Γ n'a aucune variable. Donc $\Gamma = \emptyset$ (impossible car valide) ou $\Gamma = \{\bot\}$.
- Hérédité : on montre que $\Gamma \vdash \bot$ ou bien que $\Gamma \vdash \bar{x}$ et $\Gamma \vdash x$.

Corollaire 2.1.25

Γ est insatisfaisable si et seulement si $\Gamma \vdash \bot$.

Présentation de deux algorithmes

Comment décider « systématiquement » si Γ est contradictoire ?

- L'algorithme de Davis, Putnam, Logemann et Loveland
  Parcours « intelligent » des assignations possibles des variables propositionnelles de Γ
- La Stratégie complète
  Construction des TOUTES les clauses déductibles de Γ.

Remarque

Solutions toutes les deux exponentielles en temps dans le pire des cas.

Pourquoi une complexité exponentielle ?

Rappel : on assimile une clause à l'ensemble de ses littéraux.

Exemple

Les clauses $p + \bar{q}$ et $r + q$ n'ont qu'un résolvant qui peut s'écrire :

- $p + r$
- $r + p$
- $p + r + p$
- $p + \ldots + p + r + \ldots r$

Ainsi si Γ contient n littéraux, il y a au plus 2^n clauses distinctes déduites de Γ.

Réduction d'un ensemble de clauses

Pour accélérer les algorithmes, on réduit l'ensemble de clauses.

Comment réduire ?

Enlever les clauses valides et les clauses contenant une autre clause de l'ensemble.

Exemple 2.1.27

La réduction de l'ensemble de clauses $\{p+q+\overline{p}, p+r, p+r+\overline{s}, r+q\}$ donne l'ensemble réduit :

$$\boxed{\{\cancel{p+q+\overline{p}}, p+r, \cancel{p+r+\overline{s}}, r+q\}.}$$

Justification

Propriété 2.1.28

Un ensemble de clauses E est équivalent à l'ensemble de clauses réduit obtenu à partir de E.

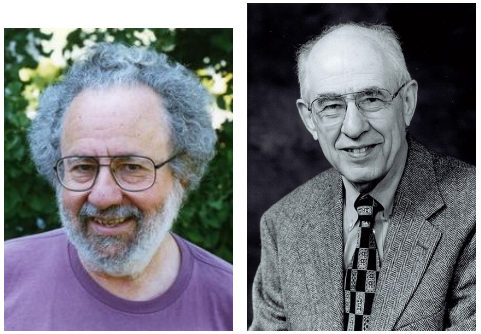
Preuve.

- On peut enlever une clause valide : $x.1 \equiv x$
- On peut enlever une clause incluant une autre clause : $x.(x+y) \equiv x$

□

Historique

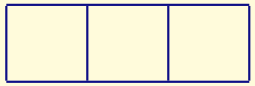
- Martin Davis (1928-), mathématicien américain
- Hilary Putnam (1926-2016), philosophe, mathématicien et informaticien américain



- règle de **résolution** (utilisation exhaustive dans l'algo initial)
- Algorithme de satisfaisabilité des formules booléennes (1960)
  - détermine un modèle d'un ensemble de clauses (si possible)
  - conçu initialement pour étudier les formules du premier ordre
  - raffinement par M. Davis, G. Logemann et D. Loveland en 1962 avec le mécanisme de branchement
  - à la base des SAT-solveurs efficaces
- Preuve d'indécidabilité des équations Diophantiennes (avec Y. Matiyasevich et J. Robinson)

Démo SAT Solveur

Problème



- Chaque case peut contenir un jeton ou pas.
- Deux cases voisines ne doivent jamais contenir un jeton toutes les deux.
- Au moins deux cases doivent contenir un jeton.

Entrées du problème : la longueur n de la grille

Modélisation booléenne

- À chaque case on associe une variable booléenne (vraie si la case contient un jeton)
- Pour le format Dimacs, on numérote les cases de 1 à n

Principe I

Deux types de transformations pour simplifier les formules :

1. préservant le sens :
   - réduction
2. préservant seulement la satisfaisabilité :
   - suppression des clauses qui ont des littéraux isolés
   - résolution unitaire

DPLL est (en général) efficace car il utilise ces deux types de transformations.

Principe II

« Branchement/Retour-arrière »

- **Branchement** : Après toutes les simplifications, affecter **vrai** à une variable bien choisie.
- Continuer récursivement l'algorithme.
- **Retour-arrière** : Si on arrive à une contradiction, on retourne au dernier choix, et on « branche » en affectant **faux** à la variable choisie.

L'algorithme DPLL (figure 2.1)

bool fonction Algo_DPLL(Γ : ensemble de clauses)

0 Supprimer les clauses valides de Γ.
Si $\Gamma = \emptyset$, renvoyer (`vrai`).
Sinon renvoyer (`DPLL`(Γ))

bool fonction `DPLL`(Γ : ensemble de clauses non valides)
La fonction renvoie vrai si et seulement si Γ est satisfaisable.

1 **Si** $\perp \in \Gamma$, renvoyer(`faux`).
Si $\Gamma = \emptyset$, renvoyer (`vrai`).
2 Réduire Γ.
3 Enlever de Γ les clauses comportant des littéraux isolés.
Si l'ensemble Γ a été modifié, aller en 1.
4 Appliquer à Γ la résolution unitaire.
Si l'ensemble Γ a été modifié, aller en 1.
5 Choisir x une variable quelconque de Γ
renvoyer (`DPLL`($\Gamma[x := 0]$) ou alors `DPLL`($\Gamma[x := 1]$))

Suppression des clauses qui ont des littéraux isolés.

Définition 2.3.1

Le littéral L est **isolé** si aucune clause de Γ ne comporte L^c.

Lemme 2.3.2

Supprimer des clauses avec un littéral isolé préserve la satisfaisabilité.

La preuve est demandée dans l'exercice 49.

Intuition : il n'y a rien à perdre à prendre $[L]_v = 1$.

Exemple 2.3.3

Soit Γ l'ensemble de clauses

(1) $p+q+r$

(2) $\bar{q}+\bar{r}$

(3) $q+s$

(4) $\bar{s}+t$

Simplifiez Γ en supprimant des clauses qui ont des littéraux isolés.

p et t sont isolés.

(2) $\bar{q}+\bar{r}$

(3) $q+s$

$\bar{r}$ et s sont maintenant isolés.
On obtient l'ensemble vide $\emptyset$.
Γ a donc un modèle (par exemple $p=1, t=1, r=0, s=1$).

Résolution unitaire

Définition 2.3.4

Une clause unitaire est une clause qui ne comporte qu'un littéral.

Lemme 2.3.5

Soit L le littéral d'une clause unitaire de Γ.
On construit un ensemble Θ :

- enlever les clauses qui comportent L
- à **l'intérieur** des clauses restantes, enlever L^c
- si Γ comporte deux clauses unitaires complémentaires, alors $\Theta = \{\bot\}$.

On applique ce procédé pour chaque clause unitaire.
Γ a un modèle si et seulement si Θ en a un.

La preuve est demandée dans l'exercice 50.

Exemple 2.3.6 Résolution unitaire

Simplifiez les ensembles de clauses suivants par résolution unitaire :

- $\Gamma = p + q,\ \bar{p},\ \bar{q}$

  $q,\ \bar{q}$ par résolution unitaire sur $\bar{p}$, puis $\perp$ par RU sur $\bar{q}$
  Donc Γ n'a pas de modèle.

- $\Gamma = a + b + \bar{d},\ \bar{a} + c + \bar{d},\ \bar{b},\ d,\ \bar{c}$

  1. $a,\ \bar{a}$
  2. $\perp$

  donc Γ n'a pas de modèle.

- $\Gamma = p,\ q,\ p + r,\ \bar{p} + r,\ q + \bar{r},\ \bar{q} + s$

  Par RU on obtient $r,\ s$.
  Donc Γ a un modèle puisque r, s en a un.

Suppression de clauses valides

Lemme 2.3.7

Soit Θ l'ensemble de clauses obtenu en supprimant les clauses valides de Γ.

Γ a un modèle si et seulement si Θ en a un.

Preuve.

$\Rightarrow$ Tout modèle de Γ est clairement un modèle de Θ car $\Theta \subset \Gamma$.

$\Leftarrow$ Supposons que Θ a un modèle v.

Si nécessaire, on le complète en assignant des valeurs quelconques aux variables présentes dans Γ mais pas dans Θ.

Toute clause de Γ est alors soit :

une clause de Θ, dont v est encore un modèle.
une clause valide, dont v est évidemment un modèle.

$\square$

L'algorithme DPLL (figure 2.1)

bool fonction Algo_DPLL(Γ : ensemble de clauses)

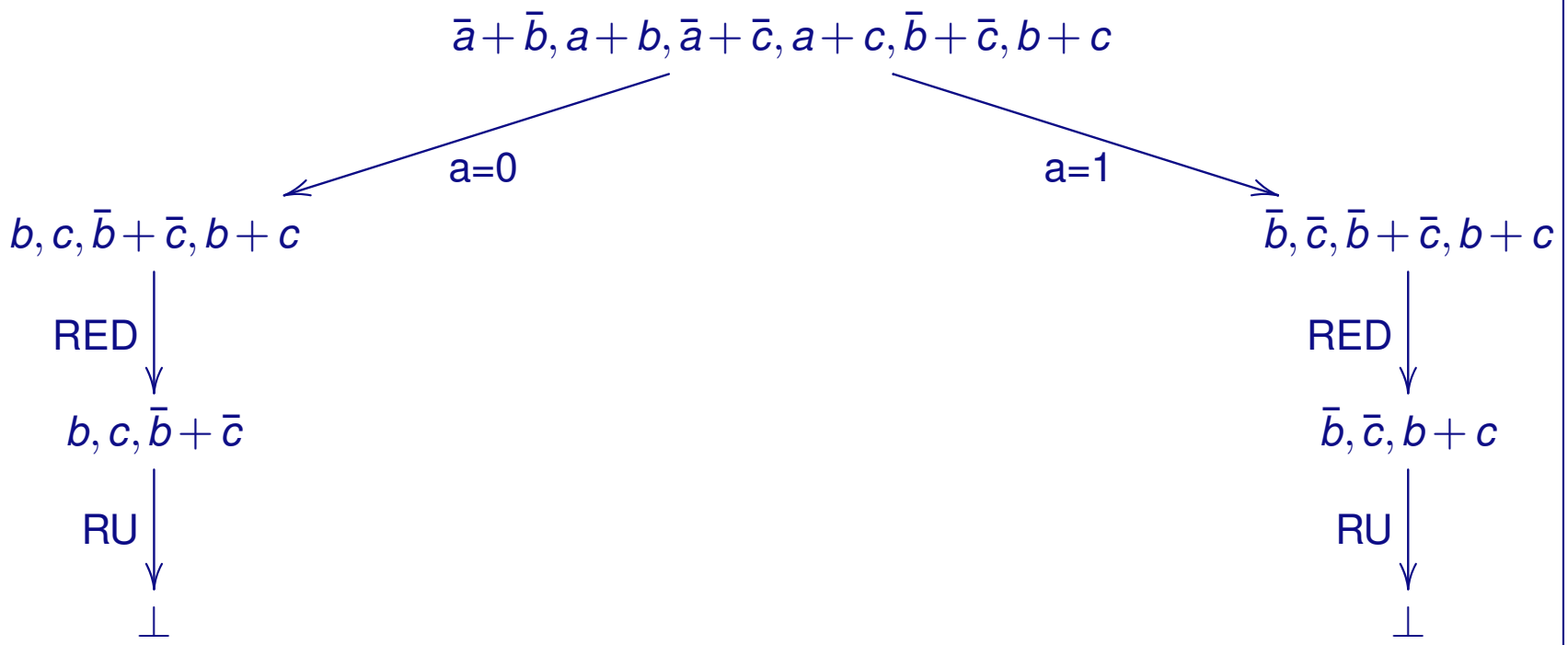
0 Supprimer les clauses valides de Γ.
Si $\Gamma = \emptyset$, renvoyer (`vrai`).
Sinon renvoyer (`DPLL`(Γ))

bool fonction `DPLL`(Γ : ensemble de clauses non valides)
La fonction renvoie vrai si et seulement si Γ est satisfaisable.

1 **Si** $\perp \in \Gamma$, renvoyer(`faux`).
Si $\Gamma = \emptyset$, renvoyer (`vrai`).
2 Réduire Γ.
3 Enlever de Γ les clauses comportant des littéraux isolés.
Si l'ensemble Γ a été modifié, aller en 1.
4 Appliquer à Γ la résolution unitaire.
Si l'ensemble Γ a été modifié, aller en 1.
5 Choisir x une variable quelconque de Γ
renvoyer (`DPLL`($\Gamma[x := 0]$) ou alors `DPLL`($\Gamma[x := 1]$))

Exemple 2.3.8

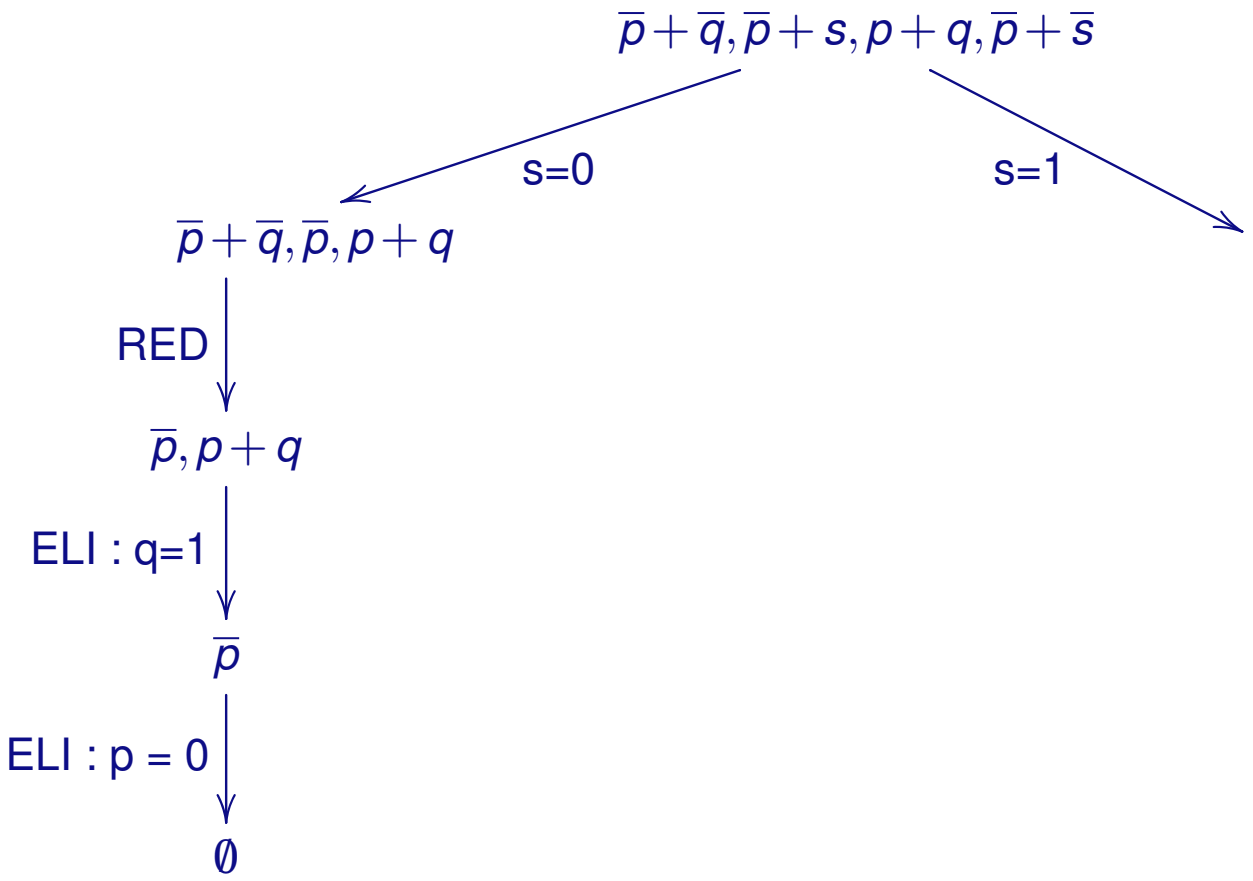
Soit Γ l'ensemble de clauses : $\bar{a}+\bar{b}, a+b, \bar{a}+\bar{c}, a+c, \bar{b}+\bar{c}, b+c$.



Puisque toutes les feuilles portent la clause vide, l'ensemble Γ est insatisfaisable.

Exemple 2.3.8

Soit Γ l'ensemble de clauses : $\overline{p}+\overline{q}, \overline{p}+s, p+q, \overline{p}+\overline{s}$.



Puisqu'une feuille porte l'ensemble vide, l'ensemble Γ est satisfaisable.
Il est inutile de poursuivre la construction de la branche droite.

Théorèmes 2.3.9 et 2.3.10

L'algorithme Algo_DPLL est correct et se termine.

Preuve de terminaison

- La suppression des clauses valides n'est exécutée qu'une seule fois.
- Boucle de simplifications : le nombre de clauses diminue strictement à chaque simplification.
- Appels récursifs : le nombre de variables diminue strictement à chaque appel récursif

D'où terminaison.

Preuve de correction

- Invariant de la boucle de simplification :
  $\boxed{\text{la valeur courante de } \Gamma \text{ a un modèle ssi } \Gamma \text{ a un modèle.}}$
  cf lemme pour chaque simplification.
- Correction des appels récursifs :
  Rappel de la propriété 2.1.21 :
  Γ *a un modèle ssi* $\Gamma[x := 0]$ *ou* $\Gamma[x := 1]$ *est satisfaisable.*
  Donc si les appels récursifs sont corrects, l'appel courant l'est aussi.

Puisque l'algorithme est correct pour un ensemble Γ sans littéraux, il est correct pour tout ensemble de clauses Γ.

Remarques 2.3.11 et 2.3.12

- **Oubli de simplifications :** DPLL reste correct si on oublie (une ou plusieurs fois) la réduction (2), l'élimination des littéraux isolés (3) et/ou la réduction unitaire (4).
- **Choix de la variable :**
  - Un bon choix pour la variable x de l'étape (5), consiste à choisir la variable qui apparait le plus souvent.
  - Un meilleur choix consiste à choisir la variable qui va entraîner par la suite le plus de simplifications.

Cf. section 2.3.5 pour les principales heuristiques de branchement.

Principe de l'algorithme : construire toutes les clauses déduites de Γ

Suivant la hauteur des arbres de preuves.

Algorithme

Pour tout entier i
Tant qu'il est possible de construire de nouvelles clauses
Construire l'ensemble réduit de toutes les clauses ayant une preuve de hauteur $\leq i$.

En pratique :
Maintenir deux suites d'ensembles de clauses, $\Delta_{i(i\geq 0)}$ et $\Theta_{i(i\geq 0)}$

Clauses minimales pour la déduction

Définition 2.1.29

On cherche à obtenir les clauses minimales pour la déduction de Γ :

- non valides
- déduites de Γ
- ne contenant aucune autre clause déduite de Γ

Exemple 2.1.30

$\Gamma = \{a + \overline{b}, b + c + d\}$

- La clause $a + c + d$ est minimale pour la déduction.
- Mais si on rajoute $\overline{a} + c$ à Γ alors $a + c + d$ n'est plus minimale (car on peut maintenant déduire $c + d$).

Propriété

Propriété 2.1.31

Soit Θ l'ensemble des clauses minimales pour la déduction de Γ.
Γ est insatisfaisable si et seulement si $\bot \in \Theta$.

Preuve.

- Supposons $\bot \in \Theta$, alors $\Gamma \vdash \bot$, donc par cohérence de la résolution, Γ est insatisfaisable.
- Supposons Γ insatisfaisable, par complétude, $\Gamma \vdash \bot$.
  Par suite $\bot$ est minimale pour la déduction de Γ, donc $\bot \in \Theta$.

$\square$

Deux suites d'ensembles de clauses

Δ_i = les nouvelles clauses utiles

Clauses déduites par une preuve de hauteur i, après élimination :

- des clauses valides
- des clauses incluant une clause de preuve de hauteur $< i$.

Δ_0 est obtenu en réduisant Γ.

Θ_i = les anciennes clauses encore utiles

Clauses déduites par une preuve de hauteur $< i$ après élimination :

- des clauses valides
- des clauses incluant une autre clause de preuve de hauteur $\leq i$.

Θ_0 est l'ensemble vide.

Construction des suites Δ_i et Θ_i

Δ_{i+1}

- Construire tous les résolvants de Δ_i et de $\Delta_i \cup \Theta_i$
- Réduire
- Enlever les résolvants incluant une clause de $\Delta_i \cup \Theta_i$

Θ_{i+1}

Enlever de $\Delta_i \cup \Theta_i$ les clauses qui incluent une clause de Δ_{i+1}.

Quand $\Delta_k = \emptyset$, arrêt de la construction :

- $k-1$ est alors la hauteur maximum d'une preuve
- Θ_k est l'ensemble réduit des clauses déduites de Γ

Exemple 2.2.1

Soit $\Gamma = \{a+b+\bar{a}, a+b, a+b+c, a+\bar{b}, \bar{a}+b, \bar{a}+\bar{b}\}$

i	Δ_i	Θ_i	$\Delta_i \cup \Theta_i$	Résolvants de Δ_i et $\Delta_i \cup \Theta_i$

i	Δ_i	Θ_i	$\Delta_i \cup \Theta_i$	Résolvants de Δ_i et $\Delta_i \cup \Theta_i$
0	$\cancel{a+b+\bar{a}}, a+b,$ $\cancel{a+b+c}, a+\bar{b}$ $\bar{a}+b, \bar{a}+\bar{b}$	$\emptyset$	$a+b, a+\bar{b},$ $\bar{a}+b, \bar{a}+\bar{b}$	$a, b, b+\bar{b},$ $a+\bar{a}, \bar{b}, \bar{a}$

i	Δ_i	Θ_i	$\Delta_i \cup \Theta_i$	Résolvants de Δ_i et $\Delta_i \cup \Theta_i$
0	$\cancel{a+b+\bar{a}}, a+b$ $\cancel{a+b+c}, a+\bar{b}$ $\bar{a}+b, \bar{a}+\bar{b}$	$\emptyset$	$a+b\cancel{a+b}, a+\bar{b}\cancel{a+\bar{b}},$ $\bar{a}+b\cancel{\bar{a}+b}, \bar{a}+\bar{b}\cancel{\bar{a}+\bar{b}}$	$a, b, \cancel{b+\bar{b}},$ $\cancel{a+\bar{a}}, \bar{b}, \bar{a}$
1	$a, b, \bar{b}, \bar{a}$	$\emptyset$	$a, b, \bar{b}, \bar{a}$	$\perp$

i	Δ_i	Θ_i	$\Delta_i \cup \Theta_i$	Résolvants de Δ_i et $\Delta_i \cup \Theta_i$
0	$\cancel{a+b+\bar{a}}, a+b$ $\cancel{a+b+c}, a+\bar{b}$ $\bar{a}+b, \bar{a}+\bar{b}$	$\emptyset$	$\cancel{a+b}, \cancel{a+\bar{b}},$ $\cancel{\bar{a}+b}, \cancel{\bar{a}+\bar{b}}$	$a, b, \cancel{b+\bar{b}},$ $\cancel{a+\bar{a}}, \bar{b}, \bar{a}$

Preuve construite

1	$a+b$	
2	$a+\overline{b}$	
3	$\overline{a}+b$	
4	$\overline{a}+\overline{b}$	
5	a	résolvant de 1 et 2
6	b	résolvant de 1 et 3
7	$\overline{b}$	résolvant de 2 et 4
8	$\overline{a}$	résolvant de 3 et 4
9	$\perp$	résolvant de 5 et 8

Exemple 2.2.2

$$\{a, c, \overline{a}+\overline{b}, \overline{c}+e\}$$

i	Δ_i	Θ_i	$\Delta_i \cup \Theta_i$	Rés. de Δ_i et $\Delta_i \cup$
0	$a, c, \overline{a}+\overline{b}, \overline{c}+e$	$\emptyset$	$a, c, \overline{a}+\overline{b}, \overline{c}+e\cancel{\overline{a}+\overline{b}, \overline{c}+e}$	$\overline{b}, e$
1	$\overline{b}, e$	a, c	$\overline{b}, e, a, c$	$\emptyset$
2	$\emptyset$	$\overline{b}, e, a, c$		

Rappel :

- $\Delta_{i+1} =$
  - Tous les résolvants de Δ_i et de $\Delta_i \cup \Theta_i$
  - Réduire
  - Enlever les résolvants qui incluent une clause de $\Delta_i \cup \Theta_i$
- $\Theta_{i+1} =$

  Enlever de $\Delta_i \cup \Theta_i$ les clauses qui incluent une clause de Δ_{i+1}

Terminaison de l'algorithme

Il y a au plus 2^n clauses déduites de Γ.

Les $\Delta_{i(i\geq 0)}$ ne comprennent que des clauses déduites de Γ.

Propriété 2.2.4

Pour tout $i \leq k$, les ensembles Δ_i sont disjoints entre eux.
(par construction de Δ_i)

Puisque les $\Delta_{i(i\geq 0)}$ sont disjoints entre eux,
il y a au plus $2^n + 1$ ensembles, d'où $k \leq 2^n + 1$.

Résultat de l'algorithme

À la terminaison de l'algorithme :

si $\bot \in \Theta_k$: Γ est insatisfaisable

si $\bot \notin \Theta_k$: Γ est satisfaisable, mais que représente Θ_k ?

- $\Theta_k =$ ensemble des clauses minimales pour la déduction.
- Γ et Θ_k sont équivalents.

Propriété 2.2.5

Pour tout $i < k$, les ensembles $\Delta_i \cup \Theta_i$ et $\Delta_{i+1} \cup \Theta_{i+1}$ sont équivalents.

D'où :

$$\Gamma \equiv \Delta_0 \cup \emptyset = \Delta_0 \cup \Theta_0 \equiv \ldots \equiv \Delta_k \cup \Theta_k = \emptyset \cup \Theta_k = \Theta_k$$

Aujourd'hui

- La résolution est un système de déduction correct et complet : il permet de **caractériser** toutes les formules insatisfaisables.
- L'**algorithme** DPLL utilise les idées de la résolution pour :
  - trouver un **modèle**
  - sinon, prouver l'**insatisfaisabilité** par une exploration efficace des assignations.
- La Stratégie Complète est un **algorithme** qui donne **toutes** les clauses déductibles d'un ensemble initial

La prochaine fois

- Déduction naturelle

À chercher : **Hypothèses** :

- (H1) : $p \Rightarrow \neg j \equiv \neg p \vee \neg j$
- (H2) : $\neg p \Rightarrow j \equiv p \vee j$
- (H3) : $j \Rightarrow m \equiv \neg j \vee m$
- ($\neg$ C) : $\neg m \wedge \neg p$ (deux clauses)

Construire la preuve de $H1, H2, H3, \neg C \vdash \bot$ obtenue par l'algorithme DPLL (ici, peu importe le choix de la variable)

Déduction Naturelle

Benjamin Wack

Université Grenoble Alpes

Février 2023

Au dernier cours

- Complétude et correction de la résolution
- Stratégie Complète
- Davis, Putnam, Logemann et Loveland

Exemple du cours : solution avec DPLL

$$\overline{p}+\overline{j}, p+j, \overline{j}+m, \overline{m}, \overline{p}$$

$$\Big\downarrow \text{RED}$$

$$p+j, \overline{j}+m, \overline{m}, \overline{p}$$

$$\Big\downarrow \text{RU : m=0, p=0}$$

$$j, \overline{j}$$

$$\Big\downarrow \text{RU}$$

$$\bot$$

Plan

Introduction à la déduction naturelle

Règles

Preuves en déduction naturelle

Conclusion

Intuition

Quand on demande une preuve dans les cours de mathématiques,

quand on décompose un raisonnement en pas élémentaires évidents,

on pratique la déduction naturelle.

La déduction naturelle (DN)

Nouveaux systèmes de déduction (1934) par Gentzen (1909-1945) :

- **Déduction naturelle** :
  - on démontre des conséquences $\Gamma \vdash p$ plutôt que des tautologies
  - un seul axiome $\Gamma,\ p \vdash p$
  - des règles d'introduction et d'élimination pour chaque connecteur
- **Calcul des séquents** :
  - $\Gamma \vdash \Delta$ si quand tout Γ est vrai, une des formules de Δ est vraie
  - règles d'introduction à gauche et à droite
  - règle de *coupure* $\dfrac{\Gamma \vdash \Delta, p \quad \Gamma', p \vdash \Delta'}{\Gamma, \Gamma' \vdash \Delta, \Delta'}$
- Calculer avec des démonstrations : **élimination des coupures**
  Toute démonstration qui n'utilise pas la règle du tiers exclus peut être transformée en une démonstration **constructive**.

Résolution *vs.* Déduction naturelle

Une preuve par **résolution** est une liste de clauses construites en utilisant n'importe quelles clauses précédentes.

En **déduction naturelle**, au cours d'une preuve, on peut ajouter des hypothèses et les enlever.

Connecteurs dérivés

$\top$, la négation et l'équivalence sont des abréviations ainsi définies :

- $\top$ abrège $\bot \Rightarrow \bot$.
- $\neg A$ abrège $A \Rightarrow \bot$.
- $A \Leftrightarrow B$ abrège $(A \Rightarrow B) \wedge (B \Rightarrow A)$.

Deux formules seront considérées comme égales si elles sont identiques aux abréviations près.

Par exemple, les formules $\neg\neg a$, $\neg a \Rightarrow \bot$ et $(a \Rightarrow \bot) \Rightarrow \bot$ sont égales.

Deux formules égales aux abréviations près sont équivalentes.

Règle

Définition 3.1.1

Une règle est constituée :

- d'**hypothèses** ou **prémisses** $H_1, \ldots, H_n$
- d'une **conclusion** C
- éventuellement on étiquette une règle par son nom (R)

$$\frac{H_1 \ldots H_n}{C}\ R$$

Exemple : Preuve d'une conjonction

$$\frac{A \quad B}{A \wedge B}\ (\wedge I)$$

Types de règles

- **Des règles d'introduction** pour introduire un connecteur dans la conclusion.
- **Des règles d'élimination** pour éliminer un connecteur d'une prémisse.
- $+$ **deux règles spéciales**

Listes des règles (système NK de Gentzen)

Table 3.1

	Introduction	Élimination
Implication	$\dfrac{\begin{array}{c}[A]\\ \vdots \\ B\end{array}}{A \Rightarrow B} \Rightarrow I$	$\dfrac{A \quad A \Rightarrow B}{B} \Rightarrow E$
Conjonction	$\dfrac{A \quad B}{A \wedge B} \wedge I$	$\dfrac{A \wedge B}{A} \wedge E1$ $\dfrac{A \wedge B}{B} \wedge E2$
Disjonction	$\dfrac{A}{A \vee B} \vee I1$ $\dfrac{B}{A \vee B} \vee I2$	$\dfrac{A \vee B \quad A \Rightarrow C \quad B \Rightarrow C}{C} \vee E$
	Règle du faux	
$\perp$	$\dfrac{\perp}{A} Efq$	
	Règle de réduction à l'absurde	
	$\dfrac{\neg\neg A}{A} RAA$	

$[A]$ signifie que A est une hypothèse

Un exemple « simple »

$$\dfrac{\dfrac{A \qquad A \Rightarrow B}{B}\Rightarrow E \qquad \dfrac{A \qquad A \Rightarrow C}{C}\Rightarrow E}{B \wedge C}\wedge I$$

Qu'a-t-on démontré ? $B \wedge C$

sous les hypothèses $A,\ A \Rightarrow B,\ A \Rightarrow C$

autrement dit $A,\ A \Rightarrow B,\ A \Rightarrow C \vDash B \wedge C$

La règle fondamentale de la Déduction Naturelle

La règle d'introduction de l'implication :

Si l'on peut déduire une formule B d'une hypothèse A, alors on peut déduire $A \Rightarrow B$ en se passant de cette hypothèse.

$$(\text{Si } A \models B \text{ alors } \models A \Rightarrow B)$$

$$\dfrac{\begin{array}{c}[A] \quad H_1 \quad \dots \quad H_n \\ \ddots \quad \vdots \quad \iddots \\ B\end{array}}{A \Rightarrow B} \Rightarrow I \qquad \text{prouve} \qquad H_1, \dots, H_n \vdash A \Rightarrow B.$$

Ligne de preuve

Définition 3.1.2

Une ligne de preuve est de l'une des trois formes suivantes :

- `Supposons` formule (pour ajouter une hypothèse)
- formule (obtenue à l'aide des règles)
- `Donc` formule (pour enlever la dernière hypothèse)

Ce dernier cas est la règle d'introduction de l'implication.

Exemples :

- `Supposons` $A \wedge B$
- A
- `Donc` $A \wedge B \Rightarrow A$

$$\dfrac{\dfrac{[A \wedge B]}{A}\,\wedge E}{A \wedge B \Rightarrow A}\,\Rightarrow I$$

Brouillon de preuve

Définition 3.1.3

Brouillon de preuve = suite de lignes
telle qu'il y a toujours au moins autant de `Supposons` que de `Donc`.

Exemple 3.1.4

numéro	ligne
1	`Supposons` a
2	$a \vee b$
3	`Donc` $a \Rightarrow a \vee b$
4	`Donc` $\neg a$
5	`Supposons` b

Brouillon de preuve : exemples

Où sont les brouillons ?

num	ligne
1	`Supposons` $a \wedge b$
2	b
3	$b \vee c$
4	`Donc` $a \wedge b \Rightarrow b \vee c$
5	`Donc` $\neg a$
6	`Supposons` b

num	ligne
1	`Supposons` a
2	$a \vee b$
3	`Donc` $a \Rightarrow a \vee b$
4	`Supposons` b

num	ligne
1	`Supposons` a
2	$a \vee b$
3	`Donc` $a \Rightarrow a \vee b$
4	`Supposons` b
5	`Donc` $\neg a$

Contexte (1/2)

- Le contexte est la suite des hypothèses introduites par les lignes `Supposons` et non enlevées par les lignes `Donc`.
- Un contexte par ligne d'un brouillon de preuve.

Exemple 3.1.6 :

contexte	numéro	ligne	règle
1	1	`Supposons` a	
1,2	2	`Supposons` b	
1,2	3	$a \wedge b$	$\wedge$I 1,2
1	4	`Donc` $b \Rightarrow a \wedge b$	$\Rightarrow$I 2,3
1,5	5	`Supposons` e	

Contexte (2/2)

Le contexte d'une formule représente les hypothèses qui ont permis de la déduire.

Définition 3.1.5

Formellement : on note Γ_i le contexte de la ligne i.

$\Gamma_0 = \emptyset$

Si la ligne i est :

- `Supposons` A
  alors $\Gamma_i = \Gamma_{i-1}, i$
- A
  alors $\Gamma_i = \Gamma_{i-1}$
- `Donc` A
  alors Γ_i est obtenu en otant la dernière formule de Γ_{i-1}

Exemple de contexte

Donnez **les contextes** du brouillon de preuve suivant :

contexte	numéro	ligne
1	1	`Supposons` a
1	2	$a \vee b$
	3	`Donc` $a \Rightarrow a \vee b$
4	4	`Supposons` b
	5	`Donc` b

Formules utilisables (1/2)

Définition 3.1.7

- La formule figurant sur une ligne est sa conclusion.
- La conclusion d'une ligne est utilisable tant que son contexte (les hypothèses qui ont permis de la déduire) est présent.

Exemple 3.1.8

contexte	numéro	ligne
1	1	`Supposons` a
1	2	$a \vee b$
	3	`Donc` $a \Rightarrow b$
	4	a
	5	$b \vee a$

La conclusion de la ligne 2 est utilisable sur la ligne 2, et pas au delà.

Formules utilisables (2/2)

Donnez les lignes sur lesquelles les formules 1 et 3 sont **utilisables** dans l'exemple suivant :

contexte	numéro	ligne
1	1	`Supposons` a
1,2	2	`Supposons` b
1,2	3	c
1	4	`Donc` d
1,5	5	`Supposons` e

Preuve

Définition 3.1.9

Soit Γ un ensemble de formules.
Une preuve dans l'environnement Γ est un brouillon de preuve tel que :

1. Pour toute ligne « `Donc` », la formule est $B \Rightarrow C$ où :
   - B est l'hypothèse enlevée
     (la dernière formule du contexte précédent)
   - C est une formule utilisable à la ligne précédente, ou dans Γ.
2. Pour toute ligne « A », la formule A est :
   - la conclusion d'une règle (autre que $\Rightarrow I$)
   - dont les prémisses sont utilisables à la ligne précédente, ou $\in \Gamma$.

Attention :

- Le contexte Γ_i change en cours de preuve.
- L'environnement Γ est fixe.

Preuve de formules

Définition 3.1.10

Une preuve de la formule A dans l'environnement Γ est une preuve :

- vide (lorsque A est élément de Γ)
- ou dont la dernière ligne a pour conclusion A et un contexte vide.

Nous notons :

- $\Gamma \vdash A$ le fait qu'il y a une preuve de A dans l'environnement Γ
- $\Gamma \vdash P : A$ le fait que P est une preuve de A dans Γ
- Lorsque l'environnement est vide, nous abrégeons $\emptyset \vdash A$ en $\vdash A$
- Si l'environnement n'est pas indiqué, $\Gamma = \emptyset$

Premier Exemple (exemple 3.1.11)

Prouvons $(a \Rightarrow b) \Rightarrow (\neg b \Rightarrow \neg a)$.

contexte	numéro	preuve	justification

contexte	numéro	preuve	justification
1	1	Supposons $a \Rightarrow b$	

contexte	numéro	preuve	justification
1	1	Supposons $a \Rightarrow b$	
1,2	2	Supposons $\neg b$	

contexte	numéro	preuve	justification
1	1	Supposons $a \Rightarrow b$	
1,2	2	Supposons $\neg b$	
1,2,3	3	Supposons a	

contexte	numéro	preuve	justification
1	1	Supposons $a \Rightarrow b$	
1,2	2	Supposons $\neg b$	

Preuves avec abréviations *vs.* sans abréviations

cont.	n.	preuve avec abréviation	preuve sans abréviation	just.
1	1	`Supposons` $a \Rightarrow b$	`Supposons` $a \Rightarrow b$	
1,2	2	`Supposons` $\neg b$	`Supposons` $b \Rightarrow \bot$	
1,2,3	3	`Supposons` a	`Supposons` a	
1,2,3	4	b	b	$\Rightarrow E$ 1, 3
1,2,3	5	$\bot$	$\bot$	$\Rightarrow E$ 2, 4
1,2	6	`Donc` $\neg a$	`Donc` $a \Rightarrow \bot$	$\Rightarrow I$ 3, 5
1	7	`Donc` $\neg b \Rightarrow \neg a$	`Donc` $(b \Rightarrow \bot) \Rightarrow (a \Rightarrow \bot)$	$\Rightarrow I$ 2, 6
	8	`Donc` $(a \Rightarrow b) \Rightarrow (\neg b \Rightarrow \neg a)$	`Donc` $(a \Rightarrow b) \Rightarrow ((b \Rightarrow \bot) \Rightarrow (a \Rightarrow \bot))$	$\Rightarrow I$ 1,7

Arbre (exemple 3.1.11)

$$\dfrac{\dfrac{\dfrac{\dfrac{(2)\cancel{\neg b} \qquad \dfrac{(1)\cancel{a \Rightarrow b} \quad (3)\cancel{a}}{(4)b}\Rightarrow E}{(5)\bot}\Rightarrow E}{(6)\neg a}\Rightarrow I[3]}{(7)\neg b \Rightarrow \neg a}\Rightarrow I[2]}{(8)(a \Rightarrow b) \Rightarrow (\neg b \Rightarrow \neg a)}\Rightarrow I[1]$$

contexte	numéro	preuve	justification
1	1	`Supposons` $a \Rightarrow b$	
1,2	2	`Supposons` $\neg b$	
1,2,3	3	`Supposons` a	
1,2,3	4	b	$\Rightarrow E$ 1, 3
1,2,3	5	$\bot$	$\Rightarrow E$ 2, 4
1,2	6	`Donc` $\neg a$	$\Rightarrow I$ 3, 5
1	7	`Donc` $\neg b \Rightarrow \neg a$	$\Rightarrow I$ 2, 6
	8	`Donc` $(a \Rightarrow b) \Rightarrow (\neg b \Rightarrow \neg a)$	$\Rightarrow I$ 1,7

Deuxième Exemple

Montrez que $a \wedge \neg a \Rightarrow b$.

contexte	numéro	preuve	justification

contexte	numéro	preuve	justification
1	1	Supposons $a \wedge \neg a$	

contexte	numéro	preuve	justification
1	1	Supposons $a \wedge \neg a$	
1	2	a	$\wedge E1$ 1

contexte	numéro	preuve	justification
1	1	Supposons $a \wedge \neg a$	
1	2	a	$\wedge E1$ 1
1	3	$\neg a$	$\wedge E2$ 1

contexte	numéro	preuve	justification
1	1	Supposons $a \wedge \neg a$	
1	2	a	$\wedge E1$ 1

Preuves avec abréviations *vs.* sans abréviations (2/2)

contexte	numéro	preuve avec abréviations	preuve sans abréviations	justification
1	1	`Supposons` $a \wedge \neg a$	`Supposons` $a \wedge (a \Rightarrow \bot)$	
1	2	a	a	$\wedge E1$ 1
1	3	$\neg a$	$a \Rightarrow \bot$	$\wedge E2$ 1
1	4	$\bot$	$\bot$	$\Rightarrow E$ 2,3
1	5	b	b	$Efq4$
	6	`Donc` $a \wedge \neg a \Rightarrow b$	`Donc` $a \wedge (a \Rightarrow \bot) \Rightarrow b$	$\Rightarrow I$ 1,5

Aujourd'hui

- La déduction naturelle propositionnelle traduit par un système formel les **règles de déduction** utilisées en langage courant.
- Contrairement à la résolution, une preuve s'effectue dans un contexte (ensemble des formules **supposées** à un instant donné)

La prochaine fois

- Complétude
- Correction
- Tactiques

À chercher :
Démonstration en déduction naturelle de

$$(p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j) \wedge (j \Rightarrow m) \Rightarrow m \vee p$$

Déduction Naturelle

Propriétés et tactiques

Benjamin Wack

Université Grenoble Alpes

Février 2023

Au dernier cours

Déduction naturelle

- Règles
- Contexte
- Preuves

Rappel des règles

	Introduction	Élimination
Implication	$\dfrac{\begin{array}{c}[A]\\ \dots\\ B\end{array}}{A \Rightarrow B}\ \Rightarrow I$	$\dfrac{A \quad A \Rightarrow B}{B}\ \Rightarrow E$
Conjonction	$\dfrac{A \quad B}{A \wedge B}\ \wedge I$	$\dfrac{A \wedge B}{A}\ \wedge E1$ $\dfrac{A \wedge B}{B}\ \wedge E2$
Disjonction	$\dfrac{A}{A \vee B}\ \vee I1$ $\dfrac{B}{A \vee B}\ \vee I2$	$\dfrac{A \vee B \quad A \Rightarrow C \quad B \Rightarrow C}{C}\ \vee E$
	Règle du faux	
$\bot$	$\dfrac{\bot}{A}\ Efq$	
	Règle de réduction à l'absurde	
	$\dfrac{\neg\neg A}{A}\ RAA$	

Deuxième Exemple

Montrez que $a \wedge \neg a \Rightarrow b$.

contexte	numéro	preuve	justification

contexte	numéro	preuve	justification
1	1	Supposons $a \wedge \neg a$	

contexte	numéro	preuve	justification
1	1	Supposons $a \wedge \neg a$	
1	2	a	$\wedge E1$ 1

contexte	numéro	preuve	justification
1	1	Supposons $a \wedge \neg a$	
1	2	a	$\wedge E1$ 1
1	3	$\neg a$	$\wedge E2$ 1

contexte	numéro	preuve	justification
1	1	Supposons $a \wedge \neg a$	
1	2	a	$\wedge E1$ 1

Un exemple avec environnement

Prouvez que $\neg A$ dans l'environnement $\neg(A \vee B)$

environnement			
référence		formule	
(i)		$\neg(A \vee B)$	
contexte	numero	preuve	justification

environnement			
référence		formule	
(i)		$\neg(A \vee B)$	
contexte	numero	preuve	justification
1	1	`Supposons` A	

environnement			
référence		formule	
(i)		$\neg(A \vee B)$	
contexte	numero	preuve	justification
1	1	`Supposons` A	
1	2	$A \vee B$	$\vee I$ 1, 1

Encore un exemple... (exemple 3.1.12)

Prouver que $\neg A \vee B$ dans l'environnement $A \Rightarrow B$.

environnement			
référence		formule	
(i)		$A \Rightarrow B$	
contexte	numero	preuve	justification

environnement			
référence		formule	
(i)		$A \Rightarrow B$	
contexte	numero	preuve	justification
1	1	`Supposons` $\neg(\neg A \vee B)$	

environnement			
référence		formule	
(i)		$A \Rightarrow B$	
contexte	numero	preuve	justification
1	1	`Supposons` $\neg(\neg A \vee B)$	

Arbre (exemple 3.1.12)

Donnez la représentation en arbre de la preuve précédente :

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(1)\cancel{\neg(\neg A \vee B)} \qquad \dfrac{\dfrac{\dfrac{\dfrac{(i)A \Rightarrow B \quad (2)\cancel{A}}{(3)B}\Rightarrow E}{(4)\neg A \vee B}\vee 2}{(5)\bot}\Rightarrow E}{(6)\neg A}\Rightarrow I[2]}{(7)\neg A \vee B}\vee 1}{(8)\bot}\Rightarrow E}{(9)\neg\neg(\neg A \vee B)}\Rightarrow I[1]}{(10)\neg A \vee B}RAA
$$

L'environnement est constitué des formules portées par les feuilles non-enlevées.

Intuitionnisme et constructivisme (Brouwer, 1881-1966)

Dans la suite de Poincaré, fondateur (en 1918) de la philosophie **intuitionniste** : les mathématiques doivent manipuler des objets accessibles à l'intuition.

- refus des objets infinis comme dans la théorie des ensembles
- en particulier notion de **réel constructible** = algorithme qui produit ses décimales

Exemple de preuve non constructive : supposons $P(0)$ et $\neg P(2)$.
Alors $\exists x(P(x) \wedge \neg P(x+1))$ ... mais on ne sait pas dire si $x = 0$ ou $x = 1$ est le « bon » témoin pour cette propriété.
Les règles d'introduction du $\vee$ explicitent lequel des cas est vrai : suivre le raisonnement pas à pas constitue un *algorithme* !

En revanche la règle $\dfrac{\neg\neg A}{A}$ permet de contourner cette contrainte.

L'exemple du cours

contexte	numero	preuve	justification
1	1	`Supposons` $(p \Rightarrow \neg j) \wedge (\neg p \Rightarrow j) \wedge (j \Rightarrow m)$	
1	2	$\neg p \Rightarrow j$	$\wedge E$ 1
1	3	$j \Rightarrow m$	$\wedge E$ 1
1,4	4	`Supposons` $\neg(m \vee p)$	
1,4,5	5	`Supposons` p	
1,4,5	6	$m \vee p$	$\vee I$ 5
1,4,5	7	$\perp$	$\Rightarrow E$ 4,6
1,4	8	`Donc` $\neg p$	$\Rightarrow I$ 5,7
1,4	9	j	$\Rightarrow E$ 2, 8
1,4	10	m	$\Rightarrow E$ 3, 9
1,4	11	$m \vee p$	$\vee I$ 10
1,4	12	$\perp$	$\Rightarrow E$ 4, 11
1	13	`Donc` $\neg\neg(m \vee p)$	$\Rightarrow I$ 4, 13
1	14	$m \vee p$	RAA 13

Théorème

Théorème 3.3.1

Si une formule A est déduite d'un environnement de formules Γ ($\Gamma \vdash A$) alors A est une conséquence de Γ ($\Gamma \models A$).

Toute preuve écrite dans un environnement Γ est correcte !

Preuve par récurrence sur le nombre de lignes d'une preuve P :

- On note H_i le contexte et C_i la conclusion de la i^e ligne de P.
- On montre que pour tout k on a $\Gamma, H_k \models C_k$.

D'où pour la dernière ligne n de la preuve $\Gamma \models A$.
(H_n est vide et $C_n = A$.)

Cas de base

Supposons que A est déduite de Γ par la preuve vide.

Alors A est élément de Γ.

Donc $\Gamma \models A$.

Hypothèse de récurrence

Supposons que pour toute ligne $i < k$ nous avons $\Gamma,\ H_i \models C_i$.

Montrons que $\Gamma,\ H_k \models C_k$.

Trois cas sont possibles :

- La ligne k est « `Supposons` C_k ».
- La ligne k est « `Donc` C_k ».
- La ligne k est « C_k ».

La ligne k est « `Supposons` C_k »

Alors la formule C_k est la dernière formule de H_k.

Donc $\Gamma, H_k \models C_k$.

La ligne k est « `Donc` C_k »

C_k est alors de la forme $B \Rightarrow D$ où :

- B est la dernière formule de H_{k-1} et $H_{k-1} = H_k, B$
- D est utilisable sur la ligne précédente.

Il existe une ligne $i < k$ telle que $D = C_i$ et H_i est préfixe de H_{k-1}.
Par hypothèse de récurrence $\Gamma,\ H_i \models D$.

Puisque H_i est préfixe de H_{k-1} on a $\Gamma,\ H_{k-1} \models D$
ce qui s'écrit également $\Gamma,\ H_k,\ B \models D$.
On en déduit que $\Gamma,\ H_k \models B \Rightarrow D$.

La ligne k est « C_k »

C_k est alors la conclusion d'une règle dont les prémisses sont :

- utilisables à la ligne précédente
- ou éléments de Γ.

On considère le seul cas de la règle $\wedge$I (autres cas analogues).
$C_k = D \wedge E$ et les prémisses de la règle sont D et E.

Par hypothèse de récurrence, nous avons :
$\Gamma,\ H_{k-1} \models D$ et $\Gamma,\ H_{k-1} \models E$.
Puisque la règle appliquée n'est pas $\Rightarrow I$, on a $H_{k-1} = H_k$.

Enfin $D,\ E \models D \wedge E$. Par transitivité $\Gamma,\ H_k \models C_k$.

Pour toutes les autres règles, la conclusion est également conséquence des prémisses.

Théorème

Uniquement pour les formules avec les symboles $\bot$, $\wedge$, $\vee$, $\Rightarrow$.

La complétude avec les symboles $\top$, $\neg$, $\Leftrightarrow$ est obtenue par le biais d'une formule équivalente (en développant les abréviations).

Théorème 3.4.1

Soient Γ un ensemble fini de formules et A une formule.
Si $\Gamma \models A$ alors $\Gamma \vdash A$.

Idée de la preuve

On définit une mesure m des formules et des listes de formules :

- $m(\bot) = 0$
- $m(x) = 1$
- $m(A \Rightarrow B) = 1 + m(A) + m(B)$
- $m(A \wedge B) = 1 + m(A) + m(B)$
- $m(A \vee B) = 2 + m(A) + m(B)$
- $m(\Gamma) = \sum_{A \in \Gamma} m(A)$

Récurrence

Soit $P(n)$ la propriété suivante :
Si $m(\Gamma,\ A) = n$, si $\Gamma \models A$ alors $\Gamma \vdash A$.

Pour montrer que $P(n)$ est vraie pour tout entier n, nous utilisons la récurrence « forte » :

Supposons que pour tout $i < k$, la propriété $P(i)$ est vérifiée.
Supposons que $m(\Gamma,\ A) = k$ et que $\Gamma \models A$.
Montrons que : $\Gamma \vdash A$.

Décomposition

On décompose A ou une formule de Γ pour pouvoir appliquer l'hypothèse de récurrence.

A est indécomposable si A est $\perp$ ou une variable.
Γ est indécomposable si Γ est une liste de littéraux ou comprend la formule $\perp$.

Subtilité de la démonstration :

- si on décompose une formule en ses sous-formules on ne retrouve pas toujours une règle de la déduction naturelle
- il faut donc parfois remplacer une formule par d'autres formules plus élaborées
- d'où la nécessité de définir la mesure m.

Remarque 3.4.2

La preuve de complétude est constructive, c'est-à-dire qu'elle donne un algorithme pour construire une preuve de A dans Γ.

Cependant les preuves ainsi construites peuvent être longues.

L'outil
http://teachinglogic.univ-grenoble-alpes.fr/DN/
construit des preuves plus efficaces.
Il utilise les tactiques « optimisées » présentées section 3.2, page 71.

Par exemple, pour prouver $B \vee C$:

- Essayez d'abord de prouver B.
- Si vous échouez, essayez de prouver C.
- Sinon, utilisez la tactique 10 (prouver C sous l'hypothèse $\neg B$).

Tactiques

Liste de 13 tactiques à utiliser dans l'ordre !

- Tactiques 1 à 3 : la preuve est terminée
- Tactiques 4 à 6 : preuve guidée par la formule à prouver (règles I)
- Tactiques 7 à 9 : preuve guidée par l'environnement (règles E)
- Tactiques 10 à 13 : raisonnement par l'absurde

Tactique 1 (trivial)

Si $A \in \Gamma$ alors la preuve obtenue est vide.

Tactique 2 (trivial + intro)

Si A est la conséquence d'une règle dont les prémisses sont dans Γ, alors la preuve obtenue est $\ll A \gg$.

Tactique 3 (Efq)

Si Γ comporte une contradiction, c'est-à-dire une formule B et une formule $\neg B$,
alors la preuve obtenue est $\ll \bot, A \gg$.

Tactique 4

Si $A = B \wedge C$ alors :

contexte	preuve	justification
Γ	B	$\cdots P \cdots$
Γ	C	$\cdots Q \cdots$
Γ	$B \wedge C$	$\wedge I$

Les preuves peuvent échouer (si $\Gamma \not\models A$).
Ici, si la preuve de B ou C échoue, celle de A aussi.

Par la suite on ne signale plus les cas d'échecs,
sauf s'il faut essayer une autre preuve.

Tactique 5

Si $A = B \Rightarrow C$, alors prouver C sous l'hypothèse B :

contexte	preuve	justification
Γ, B	`Supposons` B	
Γ, B	C	$\cdots P \cdots$
Γ	`Donc` $B \Rightarrow C$	$\Rightarrow I$

Tactique 6

Si $A = B \vee C$, alors prouver B :

contexte	preuve	justification
Γ	B	$\cdots P \cdots$
Γ	$B \vee C$	$\vee I1$

Si la preuve de B échoue alors prouver C :

contexte	preuve	justification
Γ	C	$\cdots P \cdots$
Γ	$B \vee C$	$\vee I2$

Si la preuve de C échoue aussi, essayer les tactiques suivantes.

Tactique 7

Si $B \wedge C$ est dans l'environnement, alors prouver A à partir des formules B, C, qui remplacent $B \wedge C$ dans l'environnement :

contexte	preuve	justification
$\Gamma, B \wedge C$	B	$\wedge E1$
$\Gamma, B \wedge C$	C	$\wedge E2$
$\Gamma, B \wedge C$	A	$\cdots P \cdots$

Tactique 8

Si $B \vee C$ est dans l'environnement, alors :

- prouver A dans l'environnement où B remplace $B \vee C$
- puis prouver A dans l'environnement où C remplace $B \vee C$:

contexte	preuve	justification
$\Gamma, B \vee C, B$	`Supposons` B	
$\Gamma, B \vee C, B$	A	$\cdots P \cdots$
$\Gamma, B \vee C$	`Donc` $B \Rightarrow A$	$\Rightarrow I$
$\Gamma, B \vee C, C$	`Supposons` C	
$\Gamma, B \vee C, C$	A	$\cdots Q \cdots$
$\Gamma, B \vee C$	`Donc` $C \Rightarrow A$	$\Rightarrow I$
$\Gamma, B \vee C$	A	$\vee E$

Tactique 9 ($\neg\vee$ dans Γ)

Si $\neg(B \vee C)$ est dans l'environnement, alors

- nous déduisons $\neg B$ par la preuve $P4$ et
- $\neg C$ par la preuve $P5$ (preuves demandées à l'exercice 59).
- Soit P la preuve de A dans l'environnement où $\neg B$, $\neg C$ remplacent la formule $\neg(B \vee C)$.

La preuve de A est « $P4$, $P5$, P ».

Tactique 10 (tiers exclus)

Si $A = B \vee C$, alors prouver C sous l'hypothèse $\neg B$: soit P la preuve obtenue.
« `Supposons` $\neg B$, P, `Donc` $\neg B \Rightarrow C$ » est une preuve de la formule $\neg B \Rightarrow C$ qui est équivalente à A.

Pour obtenir la preuve de A, il suffit d'ajouter la preuve $P1$, demandé à l'exercice 59 de A dans l'environnement $\neg B \Rightarrow C$.
La preuve obtenue de A est donc « `Supposons` $\neg B$, P, `Donc` $\neg B \Rightarrow C$, $P1$ ».

Tactique 11 ($\neg\wedge$ dans Γ)

Si $\neg(B \wedge C)$ est dans l'environnement, alors nous en déduisons $\neg B \vee \neg C$ par la preuve $P3$ demandée à l'exercice 59 puis nous raisonnons par cas comme ci-dessus :

- prouver A dans l'environnement où $\neg B$ remplace $\neg(B \wedge C)$: Soit P la preuve obtenue,
- prouver A dans l'environnement où $\neg C$ remplace $\neg(B \wedge C)$: Soit Q la preuve obtenue.

La preuve de A est « $P3$, `Supposons` $\neg B$, P, `Donc` $\neg B \Rightarrow A$, `Supposons` $\neg C$, Q, `Donc` $\neg C \Rightarrow A$, A ».

Tactique 12 ($\neg \Rightarrow$ dans Γ)

Si $\neg(B \Rightarrow C)$ est dans l'environnement, alors

- nous déduisons B par la preuve $P6$,
- $\neg C$ par la preuve $P7$ (preuves demandées à l'exercice 59).
- Soit P la preuve de A dans l'environnement où B, $\neg C$ remplacent la formule $\neg(B \Rightarrow C)$.

La preuve de A est « $P6$, $P7$, P ».

Tactique 13 ($\Rightarrow$ dans Γ)

Si $B \Rightarrow C$ est dans l'environnement et si $C \neq \bot$, autrement dit si $B \Rightarrow C$ n'est pas égale à $\neg B$, alors,
nous déduisons $\neg B \vee C$ dans l'environnement $B \Rightarrow C$ par la preuve $P2$ demandée à l'exercice 59 puis nous raisonnons par cas :

- prouver A dans l'environnement où $\neg B$ remplace $B \Rightarrow C$: Soit P la preuve obtenue,
- prouver A dans l'environnement où C remplace $B \Rightarrow C$: Soit Q la preuve obtenue.

La preuve de A est « $P2$, `Supposons` $\neg B$, P, `Donc` $\neg B \Rightarrow A$, `Supposons` C, Q, `Donc` $C \Rightarrow A$, A ».

Exemple

Preuve de la formule de Pierce :

$$((p \Rightarrow q) \Rightarrow p) \Rightarrow p$$

Plan de preuve

Tactique 5 ($\Rightarrow$I) obligatoire !

Preuve Q :
`Supposons` $(p \Rightarrow q) \Rightarrow p$
Q_1 preuve de p dans l'environnement $(p \Rightarrow q) \Rightarrow p$
`Donc` $((p \Rightarrow q) \Rightarrow p) \Rightarrow p$

La preuve Q_1 utilise nécessairement la tactique 13 (environnement formé de $B \Rightarrow C \ = \ (p \Rightarrow q) \Rightarrow p$).

Donc on doit prouver p :

- dans l'environnement $\neg B = \neg(p \Rightarrow q)$
- dans l'environnement $C = p$.

Plan de la preuve de Q_1

Preuve Q_1 :

Q_{11} est la preuve de $\neg B \vee C$ dans l'environnement $B \Rightarrow C$, voir exercice 59

`Supposons` $\neg(p \Rightarrow q)$

Q_{12} preuve de p dans l'environnement $\neg(p \Rightarrow q)$

`Donc` $\neg(p \Rightarrow q) \Rightarrow p$

`Supposons` p

Q_{13} preuve de p dans l'environnement p

`Donc` $p \Rightarrow p$

p

Preuve de Q_1

Q_{13} est la preuve vide, car $A = C = p$.

Q_{12} est la preuve de $A = p$ dans l'environnement $\neg(p \Rightarrow q)$.
(c'est la preuve P_6 demandée à l'exercice 59)

En recollant les morceaux $Q_1,\ Q_{11},\ Q_{12},\ Q_{13}$, nous obtenons la preuve Q.

On peut aussi faire la preuve Q_{12} sans utiliser les tactiques.

Aujourd'hui

- La Déduction Naturelle propositionnelle est correcte et complète.
- Tactiques pour la construction d'une preuve

Preuves automatiques

Pour produire automatiquement des preuves sous forme de tableau, on recommande d'utiliser le logiciel (il implémente les treize tactiques précédentes) :

http://teachinglogic.univ-grenoble-alpes.fr/DN/

Pour construire des preuves sous forme d'arbres, on peut s'aider du logiciel :

http://www-sop.inria.fr/marelle/Laurent.Thery/peanoware/Nd.html

Plan du Semestre

RENTREE S4

- Logique propositionnelle
- Résolution propositionnelle
- Déduction naturelle propositionnelle *
- Logique du premier ordre

PARTIEL

- Base de la démonstration automatique (« résolution au premier ordre »)
- Déduction naturelle au premier ordre

EXAMEN

Avant de commencer

À propos du partiel

- 2 heures
- une feuille A4 de notes manuscrites autorisée
- au programme : toute la logique propositionnelle
- Exercices type (cf. annales), un exo du poly

Avant le partiel

- Pensez au pré-rapport du projet !

Rappel du planning, annales... voir
https://wackb.gricad-pages.univ-grenoble-alpes.fr/inf402/

Logique du premier ordre
Première partie :
Langage et Sens des Formules

Benjamin Wack

Université Grenoble Alpes

Mars 2023

Plan de l'UE

- Logique propositionnelle : $\wedge, \vee, \neg, \Rightarrow, \Leftrightarrow$
- Interprétation : fonctions booléennes
- Systèmes de déduction : résolution, déduction naturelle
- Algorithmes : stratégie complète, DPLL, tactiques en DN

- Logique du premier ordre : $\forall, \exists$
- Interprétation
- « Résolution au premier ordre »
- Déduction naturelle au premier ordre

Plan

Introduction

Langage
- Formules (strictes)
- Formule à priorités

Être libre ou lié

Sens des symboles
- Déclaration de symbole
- Signature

Interprétation

Interprétation finie

Conclusion

Aperçu de la logique du premier ordre

Un domaine : les *objets* sur lesquels on raisonne

Trois catégories :

- les termes qui représentent des éléments du domaine
- les relations entre éléments du domaine
- les formules qui décrivent les interactions entre les relations

Deux nouveaux symboles (quantificateurs) dans les formules

$\forall$ (quantificateur universel) et $\exists$ (quantificateur existentiel)

Exemples :

- domaine = membres d'une famille
- le terme $pere(x)$ désigne un élément du domaine (le père de x)
- la relation *frere* détermine si deux éléments sont frères
- la formule $\forall x \exists y\ frere(y, x)$ signifie "tout individu a un frère".

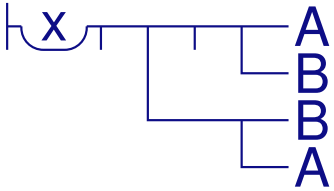
Syllogisme

Tous les hommes sont mortels.
Socrate est un homme.
Donc Socrate est mortel.

$$\forall x(homme(x) \Rightarrow mortel(x))$$
$$homme(Socrate)$$
$$mortel(Socrate)$$

Begriffsschrift (idéographie) de Gottlob Frege (1879)

- Dans la lignée de Leibniz, langage formel « universel »



- *Système* logique du premier ordre
  (contient certaines règles comme le Modus Ponens déjà connues des Stoïciens, mais aussi de nouvelles règles pour les quantificateurs)
- Ne contient que des règles de *raisonnement* mais permettent d'exprimer toutes les *notions mathématiques* (à partir des ensembles)
- Contient également la logique du **second ordre** :
  une variable peut représenter une propriété $\forall R \exists x R(x)$

Plan de la séance

Introduction

Langage
- Formules (strictes)
- Formule à priorités

Être libre ou lié

Sens des symboles
- Déclaration de symbole
- Signature

Interprétation

Interprétation finie

Conclusion

Vocabulaire

- Deux constantes propositionnelles : $\bot$ et $\top$
- Connecteurs : $\neg, \wedge, \vee, \Rightarrow, \Leftrightarrow$
- Quantificateurs : **universel** $\forall$ et **existentiel** $\exists$
- Variables : $u, v, w, x, y, z, x1, x2 \ldots$ ($\neq$ var. propositionnelles)
- Symboles : $a, b, c, p, frere, 12 \ldots$
- Ponctuations : virgule et parenthèses (pour noter les arguments d'un symbole)

Exemple 4.1.1

- $x, x1, x2, y$ sont des variables,
- $f1$, *succ*, *somme*, 12, 24, *homme*, *frere*, *ilPleut* sont des symboles :
  - des **fonctions** à un, plusieurs arguments ou aucun (constantes)
  - des **relations** à un, plusieurs arguments (relations) ou aucun (variables propositionnelles)
- Pour certains symboles (dits *spéciaux*)
  on s'autorisera à écrire par exemple $x = y$ ou $z > 3$.

Terme

Définition 4.1.2

Un terme peut être :

- un symbole s seul
- ou une variable x
- ou un symbole appliqué à des termes $s(t_1, \ldots, t_n)$

Exemple 4.1.3

x ; a ; $f(x1, x2, g(y))$; $somme(5, produit(x, 42))$ sont des termes.

Par contre $f(\perp, 2, y)$ n'est pas un terme.

Notons que $42(1, y, 3)$ est aussi un terme, mais on utilise rarement 42 pour un nom de fonction ou de relation.

Formule atomique

Définition 4.1.4

Une formule atomique est :

- $\top$ ou $\bot$
- ou un symbole seul
- ou un symbole appliqué à des termes $s(t_1, \ldots, t_n)$

Exemple 4.1.5 :

- $P(x)$, a et $R(1, +(5, 42), g(z))$ sont des formules atomiques
- x et $A \vee f(4, 2, 6)$ ne sont pas des formules atomiques

Attention : Interprétation à deux niveaux

L'ensemble des **termes** et l'ensemble des **formules atomiques** ne sont pas disjoints.

Par exemple $p(x)$ peut être à la fois un terme **et** une formule atomique.

- $[\![t]\!]$ désigne le sens de t vu comme un terme.
- $[t]$ désigne le sens de t vu comme une formule.

Formule (stricte)

Définition 4.1.6

Une formule (stricte) est :

- une formule atomique
- $\neg A$

  avec A une formule
- $(A \circ B)$

  avec A et B des formules et $\circ$ un connecteur $\vee, \wedge, \Rightarrow, \Leftrightarrow$
- $\forall x\ A$ ou $\exists x\ A$

  avec A une formule et x une variable quelconque

Exemple 4.1.7

- $\boxed{homme(x)\ ;\ \ parent(fils(y), mere(Alice))\ ;\ \ = (x, +(f(x), g(y)))}$

  sont des formules atomiques, donc des formules.
- Par contre

  $\boxed{\forall x\ (homme(x) \Rightarrow homme(Socrate))}$

  est une formule qui n'est pas atomique.

Formule (stricte) : Exemples

Parmi ces expressions, lesquelles sont des formules strictes :

- x
  pas une formule
- a
  oui
- $(a(x) \Rightarrow b) \wedge a(x) \Rightarrow b$
  non : il manque des ()
- $\exists x((\bot \Rightarrow a(x)) \wedge b(x))$
  oui
- $\exists x \exists y < (-(x, y), +(a, y))$
  oui

Abréger l'écriture

Écriture infixée : les fonctions $+, -, *, /$ et les relations $=, \neq, <, >, \leq, \geq$ s'écrivent de manière usuelle.

Exemple 4.1.9

- $\leq (*(3, x), +(y, 5))$ s'**abrège** en $3 * x \leq y + 5$
- $+(x, *(y, z))$ s'**abrège** en $x + y * z$

Formule à priorités

Définition 4.1.10

Une formule à priorité est :

- Une formule atomique.
- $\neg A$
- $A \circ B$ pour un connecteur binaire $\circ$
- $\forall x\ A$ ou $\exists x\ A$
- (A)

Pour interpréter une formule abrégée

Donner des priorités

- On conserve les priorités choisies sur les connecteurs booléens.
- La priorité des quantificateurs est identique à celle de la négation.
- Les connecteurs sont moins prioritaires que les relations

Tableau 4.1 récapitulatif des priorités

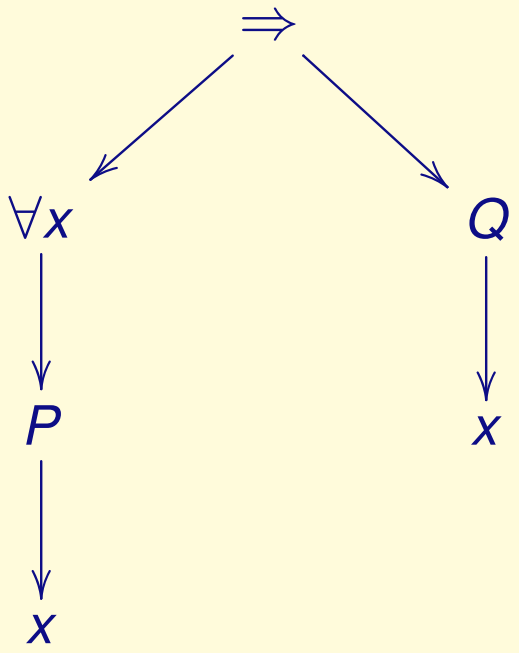
Priorités décroissantes du haut vers le bas.

OPÉRATIONS	
$*, /, +, -$, etc.	
RELATIONS	
$=, \neq, <, \leq, >, \geq$, etc.	
NÉGATION, QUANTIFICATEURS	
$\neg, \forall, \exists$	
CONNECTEURS BINAIRES	
$\wedge$	associatif gauche
$\vee$	associatif gauche
$\Rightarrow$	associatif **droit**
$\Leftrightarrow$	associatif gauche

Représentation en arbre

Exemple 4.1.12 $\forall x P(x) \Rightarrow Q(x)$

$\forall$ est prioritaire : l'opérande gauche de l'implication est $\forall x P(x)$.



Plan de la séance

Introduction

Langage
- Formules (strictes)
- Formule à priorités

Être libre ou lié

Sens des symboles
- Déclaration de symbole
- Signature

Interprétation

Interprétation finie

Conclusion

Idée

- Le sens de la formule $x+2=4$ dépend de x.
  La formule n'est vraie (en arithmétique) que si $x=2$.
  x est libre dans cette formule
- $\forall x(x+2=4)$ est insatisfaisable (en arithmétique).
  $\forall x(x+0=x)$ est valide.
  Il n'y a pas à choisir de valeur pour x.
  Ces deux formules n'ont pas de variables libres.
- Le nom de la variable n'a alors plus d'importance.
  Situation courante en mathématiques $\int_0^1 f(x)dx$
  ... et en informatique

```
int Toto(int x) {
  return x + 1;
}
```

Occurences libres et liées

Définition 4.2.1

Un quantificateur lie une variable localement.

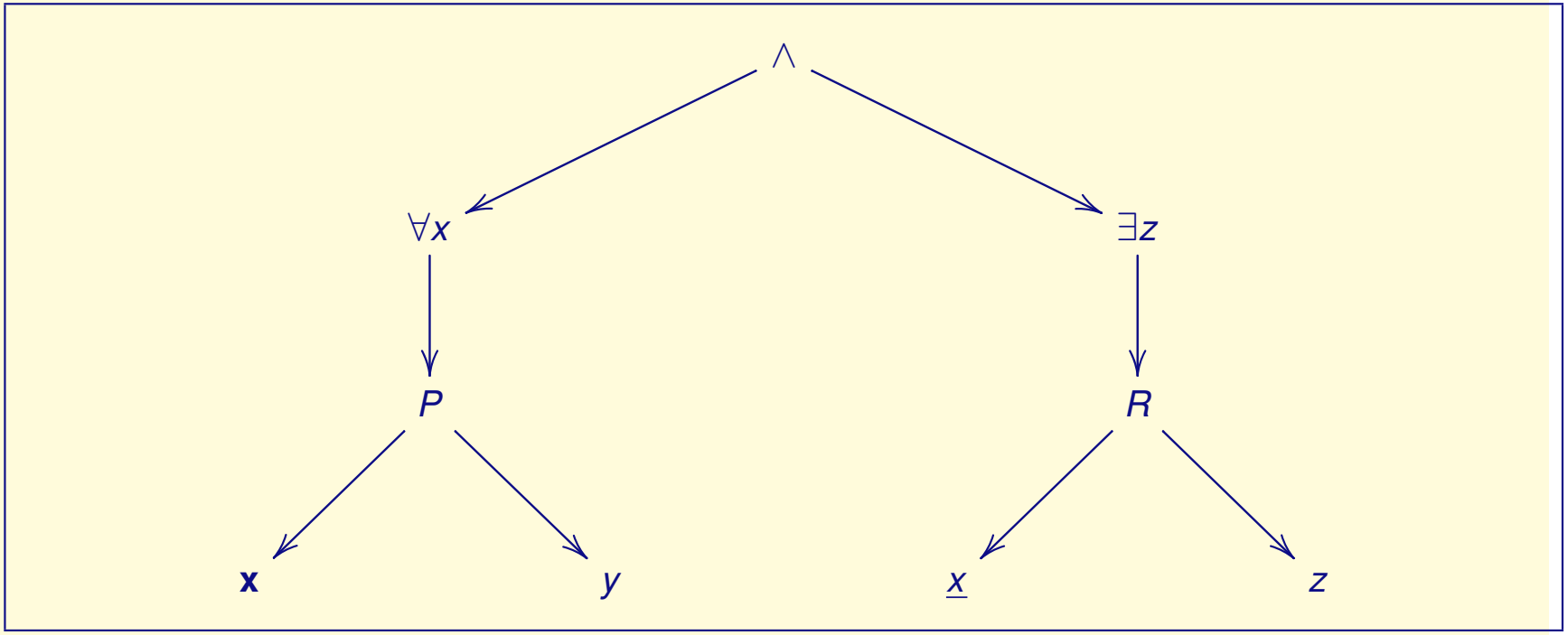
- Dans $\forall x\ A$ ou $\exists x\ A$, la portée de la liaison pour x est A.
- Une occurrence de x est liée si elle est dans la portée d'une liaison pour x.
- Sinon elle est dite libre.

Si nous représentons une formule par un arbre :

- Une occurrence de x est liée si elle est en dessous d'un nœud $\exists x$ ou $\forall x$.
- Toute autre occurrence de x est libre.

Exemple 4.2.2

$\forall x P(\mathbf{x}, y) \wedge \exists z R(\underline{x}, z)$



- L'occurrence de z est liée et celle de y est libre.
- L'occurrence en gras de x est liée.
- L'occurrence soulignée de x est libre.

Variables libres, liées

Définition 4.2.3

- Une formule sans variable libre est dite formule fermée.

Remarque

- Dans $\forall x P(x) \vee Q(x)$, la variable x est à la fois libre et liée (donc la formule n'est pas fermée).

Exemple 4.2.6

Les variables libres de $\forall x P(x, y) \wedge \exists z R(x, z)$ sont x et y.

Plan de la séance

Introduction

Langage
- Formules (strictes)
- Formule à priorités

Être libre ou lié

Sens des symboles
- Déclaration de symbole
- Signature

Interprétation

Interprétation finie

Conclusion

Déclaration de symbole

Définition 4.3.1

Une déclaration de symbole est un triplet noté s^{gn} où :

- s est un symbole
- g est une des lettres f (signifiant fonction) ou r (signifiant relation)
- n est un entier naturel.

Remarque 4.3.3

g et n sont facultatifs s'ils sont clairs étant donné le contexte.

Exemple : **égal** est toujours une relation à 2 arguments.
On écrit donc $=$ au lieu de $=^{r2}$.

Déclaration de symbole : Exemple

Exemple 4.3.2

- $frere^{r2}$ est une **(r)elation** avec **2** arguments
- $*^{f2}$ est une **(f)onction** avec **2** arguments
- $homme^{r1}$ est une **relation** unaire

Signature

Définition 4.3.4

Une signature Σ est un ensemble de déclarations de symboles.

Selon la déclaration du symbole s, on dira qu'il est :

1. pour s^{fn} : un symbole de fonction à n arguments
2. pour s^{f0} : une constante
3. pour s^{rn} : une relation à n arguments
4. pour s^{r0} : une variable propositionnelle

Exemple en mathématiques

Définissons une signature pour l'arithmétique :

- Constantes $0^{f0}, 1^{f0}$
- Fonctions $+^{f2}, -^{f2}, *^{f2}$
- Relations $=^{r2}$

Remarques :

- Le contexte étant clair, on écrira plutôt 0, 1, $+$, $-$, $*$ et $=$.
- On peut cependant préciser que $-$ attend deux arguments (car il existe en version unaire).

Relation unaire : une relation à 1 argument est simplement une propriété d'un terme (par exemple ici $premier^{r1}$).

Terme sur une signature

Définition 4.3.8

Un terme sur Σ est :

- une variable,
- ou une constante s^{f0}
- ou un terme de la forme $s(t_1, \ldots, t_n)$ avec
  - s^{fn}
  - $n \geq 1$
  - $t_1, \ldots, t_n$ des termes sur Σ

Formule atomique sur une signature

Définition 4.3.9

Une formule atomique sur Σ est :

- une constante $\top$ ou $\bot$
- ou une variable propositionnelle s^{r0}
- ou une expression de la forme $s(t_1, \ldots, t_n)$ avec
  - s^{rn}
  - $n \geq 1$
  - $t_1, \ldots, t_n$ des **termes** sur Σ

Formule sur une signature

Définition 4.3.10

Une formule sur Σ est une formule dont les sous-formules atomiques sont correctes pour Σ.

Exemple 4.3.11

$\forall x\ (p(x) \Rightarrow \exists y\ q(x,y))$ est une formule sur $\Sigma = \{p^{r1}, q^{r2}, h^{f1}, c^{f0}\}$.

Mais c'est aussi une formule sur $\Sigma' = \{p^{r1}, q^{r2}\}$
(les symboles h et c ne figurent pas dans la formule).

La signature associée à une formule est la plus petite signature Σ qui convient pour cette formule.

Plan de la séance

Introduction

Langage
- Formules (strictes)
- Formule à priorités

Être libre ou lié

Sens des symboles
- Déclaration de symbole
- Signature

Interprétation

Interprétation finie

Conclusion

Interprétation

Définition 4.3.16

Une interprétation I sur une signature Σ est définie par :

- un domaine D non vide
- à chaque symbole s^{gn} on associe sa valeur s_I^{gn} comme suit :

(constante)	s_I^{f0} est un élément de D
(fonction)	s_I^{fn} est une fonction de $D^n \to D$
(variable propositionnelle)	s_I^{r0} vaut 0 ou 1
(relation)	s_I^{rn} est un ensemble de n-uplets dans D (ceux qui vérifient cette relation)

Exemple 4.3.17

Soit une relation binaire *ami* et le domaine $D = \{1, 2, 3\}$.
On considère l'interprétation I donnée par
$ami_I^{r2} = \{(1, 2), (1, 3), (2, 3)\}$.

Dans cette interprétation, on dira que la formule $ami(2, 3)$ est vraie mais que la formule $ami(2, 1)$ est fausse.

Remarque 4.3.18

Dans toute interprétation I,
la valeur du symbole $=$ est l'ensemble $\{(d, d) \mid d \in D\}$,
autrement dit le sens de l'égalité est l'identité sur le domaine D.

État, assignation

Une interprétation définit seulement le sens de la signature (les symboles), pas celui des variables ni des formules.

Définition 4.3.21

Un état e d'une interprétation associe à chaque variable un élément du domaine D.

Définition 4.3.22

Une assignation est un couple (I, e) composé d'une interprétation I et d'un état e.

Exemple 4.3.23

Soient le domaine $D = \{1,2,3\}$ et l'interprétation I
$ami_I^{r2} = \{(1,2),(1,3),(2,3)\}$

L'interprétation I ne suffit pas à déterminer le sens de $ami(x,y)$.

Soit e l'état qui associe 2 à x et 1 à y.

L'assignation (I,e) rend la formule $ami(x,y)$ fausse.

Exemple 4.3.31

Soit l'interprétation I de domaine $D = \{1,2,3\}$ donnée par $ami_I^{r2} = \{(1,2),(1,3),(2,3)\}$

Comment interpréter la formule $ami(1,2) \wedge ami(2,3) \Rightarrow ami(1,3)$ dans I ?
On sait interpréter les formules atomiques :

- $[ami(1,2)]_I = vrai$
- $[ami(2,3)]_I = vrai$
- $[ami(1,3)]_I = vrai$

On peut alors procéder comme d'habitude avec les connecteurs, d'où $[ami(1,2) \wedge ami(2,3) \Rightarrow ami(1,3)]_I = vrai$:
Cette formule est vraie dans l'interprétation I.

Plan de la séance

Introduction

Langage
- Formules (strictes)
- Formule à priorités

Être libre ou lié

Sens des symboles
- Déclaration de symbole
- Signature

Interprétation

Interprétation finie

Conclusion

Modèle fini

Définition

Un modèle fini d'une formule fermée est une interprétation de la formule de *domaine fini*, qui rend vraie la formule.

Remarque

- Le nom des éléments du domaine est sans importance.
- Ainsi pour un modèle avec n éléments, nous utiliserons le domaine des entiers naturels inférieurs à n.

Construire un modèle fini

Idée naïve : Pour savoir si une formule fermée a un modèle de domaine $\{0, \ldots, n-1\}$, il suffit de

- **énumérer** toutes les interprétations possibles de la signature associée à la formule
- **évaluer** la formule pour chacune de ces interprétations.

Exemple

Soit $\Sigma = \{a^{f0}, f^{f1}, P^{r2}\}$.

Sur un domaine à 5 éléments, Σ a $5 \times 5^5 \times 2^{25}$ interprétations !

Cette méthode est inutilisable en pratique.

Méthode pour la recherche d'un modèle fini

Recherche de modèles à n éléments **par réduction au cas propositionnel**

Cas simple : formule n'ayant ni symbole de fonction ni constante.

Construction du modèle à n éléments

1. Suppression des quantificateurs : remplacer A par sa n-expansion B.
2. Dans B,
   - remplacer les égalités par leur valeur ($i = j$ est vrai *ssi* i et j sont identiques)
   - Appliquer les simplifications usuelles

   Soit C la formule ainsi obtenue.
3. Chercher un modèle de C sous la forme d'une assignation propositionnelle des formules atomiques de C.

Expansion d'une formule

Définition 4.3.39

La n-expansion de A est obtenue en remplaçant :

- toute sous-formule de A de la forme $\forall x B$ par la conjonction $\bigwedge_{i<n} B < x := i >$
- toute sous-formule de A de la forme $\exists x B$ par la disjonction $\bigvee_{i<n} B < x := i >$

Exemple 4.3.40

La 2-expansion de la formule $\exists x P(x) \Rightarrow \forall x P(x)$ est la formule

$$\boxed{P(0) \vee P(1) \Rightarrow P(0) \wedge P(1)}$$

Exemple 4.3.45

$A \;=\; \exists x P(x) \wedge \exists x \neg P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y))$

A n'a pas de modèle à un élément.
$(P(0) \wedge \neg P(0) \wedge (P(0) \wedge P(0) \Rightarrow 0 = 0)$ est insatisfaisable.)

2-expansion de A

$$(P(0)+P(1)).\ \ (\overline{P(0)}+\overline{P(1)}).\quad (P(0).P(0) \Rightarrow 0=0).(P(0).P(1) \Rightarrow 0=1).$$
$$(P(0)+P(1)).\ (\overline{P(0)}+\overline{P(1)}).\quad (P(1).P(0) \Rightarrow 1=0).(P(1).P(1) \Rightarrow 1=1)$$

En remplaçant les égalités par leur valeurs

$$(P(0)+P(1)).\ \ (\overline{P(0)}+\overline{P(1)}).$$
$$(P(0).P(0) \Rightarrow \top).\ (P(0).P(1) \Rightarrow \bot).\ (P(1).P(0) \Rightarrow \bot).\ (P(1).P(1) \Rightarrow \top)$$

Ce qui se simplifie en $(P(0)+P(1)).(\overline{P(0)}+\overline{P(1)})$
L'assignation $P(0) = \mathit{vrai}, P(1) = \mathit{faux}$ en est un modèle propositionnel, donc l'interprétation I de domaine $\{0,1\}$ où $P_I = \{0\}$ est modèle de A.

Logiciel pour construire un modèle fini

MACE

- **traduction** des formules du premier ordre en formules propositionnelles
- **algorithmes performants pour trouver la satisfaisabilité** d'une formule propositionnelle (par exemple `DPLL`)

`http://www.cs.unm.edu/~mccune/mace4`

Un exemple concret :
`http://www.cs.unm.edu/~mccune/mace4/examples/2009-11A/mace4-misc/`

Plan de la séance

Introduction

Langage
- Formules (strictes)
- Formule à priorités

Être libre ou lié

Sens des symboles
- Déclaration de symbole
- Signature

Interprétation

Interprétation finie

Conclusion

Aujourd'hui

- La **logique du premier ordre** est une logique pourvue de quantificateurs $\forall$ et $\exists$
- Ces quantificateurs portent sur des variables qui représentent des éléments d'un domaine
- Les **formules atomiques** sont construites à l'aide de symboles de fonctions et de relations entre éléments du domaine
- Pour interpréter une formule :
  - Les symboles doivent être interprétés dans un domaine
  - Les variables libres doivent être évaluées à l'aide d'un état
- Méthode de recherche d'un **(contre-)modèle** par interprétation finie et expansion

La prochaine fois

- Interprétation d'une formule du premier ordre
- Notion de modèle
- Équivalences remarquables

À chercher : traduire en logique du premier ordre

- Il y a des gens qui s'aiment.
- Si deux personnes s'aiment, alors elles sont conjointes.
- On ne peut pas aimer deux personnes à la fois.

Logique du premier ordre
Deuxième partie :
Interprétation d'une formule

Benjamin Wack

Université Grenoble Alpes

Mars 2023

Quelques exemples

Traduire en logique du premier ordre :

- Il y a des gens qui s'aiment.

  $\exists x\ \exists y\ (a(x,y) \wedge a(y,x))$

- Si deux personnes s'aiment l'une l'autre, alors elles sont conjointes.

  $\forall x\ \forall y\ (a(x,y) \wedge a(y,x) \Rightarrow c(x) = y \wedge c(y) = x)$

- On ne peut pas aimer deux personnes à la fois.

  $\forall x\ \forall y\ (a(x,y) \Rightarrow \forall z\ (a(x,z) \Rightarrow y = z))$

  $\forall x\ \forall y\ \forall z\ (a(x,y) \wedge a(x,z) \Rightarrow y = z)$

Plan

Sens des formules

Interprétation finie par expansion : compléments

Interprétation et substitution

Equivalences remarquables

Conclusion

Plan

Sens des formules

Interprétation finie par expansion : compléments

Interprétation et substitution

Equivalences remarquables

Conclusion

Rappel : Interprétation et état

Définition 4.3.16

Une interprétation I sur une signature Σ est définie par :

- un domaine D non vide
- pour chaque symbole $s^{gn} \in \Sigma$:

(constante)	s_I^{f0} est un élément de D
(fonction)	s_I^{fn} est une fonction de $D^n \to D$
(variable propositionnelle)	s_I^{r0} vaut 0 ou 1
(relation)	s_I^{rn} est un sous-ensemble de D^n

Définition 4.3.21

Un état e associe à chaque variable un élément du domaine D.

Remarque 4.3.24

- Pour une formule avec des variables libres, nous avons besoin d'une assignation (I, e) dont l'état est précisé.
- Pour une formule sans variable libre, il suffit de donner une interprétation I des symboles de la formule.

  En effet (I, e) et (I, e') donneront la même valeur à toutes les formules :
  on assimile donc (I, e) et I.

Termes

Définition 4.3.25 Évaluation

Définition inductive :

1. si t est une variable, alors $[\![t]\!]_{(I,e)} = e(t)$
2. si t est une constante alors $[\![t]\!]_{(I,e)} = t_I^{f0}$
3. si $t = s(t_1, \ldots, t_n)$ avec s un symbole de fonction, alors $[\![t]\!]_{(I,e)} = s_I^{fn}([\![t_1]\!]_{(I,e)}, \ldots, [\![t_n]\!]_{(I,e)})$

Exemple 4.3.26

Soit la signature a^{f0}, f^{f2}, g^{f2}.
Soit I l'interprétation de domaine $\mathbb{N}$ dans laquelle :

- a est interprété par l'entier 1 ;
- f est interprétée comme le produit ;
- g est interprétée comme la somme.

Soit e l'état tel que $e(x) = 2$ and $e(y) = 3$.
Calculons $[\![f(x, g(y, a))]\!]_{(I,e)}$.

$$\begin{array}{rcl} [\![f(x,g(y,a))]\!]_{(I,e)} & = & [\![x]\!]_{(I,e)} * [\![g(y,a)]\!]_{(I,e)} \\ & = & [\![x]\!]_{(I,e)} * ([\![y]\!]_{(I,e)} + [\![a]\!]_{(I,e)}) \\ & = & e(x) * (e(y) + 1) \\ & = & 2 * (3 + 1) = 8 \end{array}$$

Formules

Définition 4.3.27 Sens des formules atomiques

Une formule atomique est de l'une des trois formes suivantes :

1. Une constante propositionnelle : $[\top]_{(I,e)} = 1$ et $[\bot]_{(I,e)} = 0$
2. Une variable propositionnelle : $[s]_{(I,e)} = s_I^{r0}$
3. Un terme $A = s(t_1, \ldots, t_n)$ avec s un symbole de relation :
   - si $(\llbracket t_1 \rrbracket_{(I,e)}, \ldots, \llbracket t_n \rrbracket_{(I,e)}) \in s_I^{rn}$ alors $[A]_{(I,e)} = 1$
   - sinon $[A]_{(I,e)} = 0$

Exemple 4.3.19

On considère la signature suivante :

- $Anne^{f0}$, $Bernard^{f0}$ et $Claude^{f0}$: constantes
- a^{r2} : relation à deux arguments ($a(x,y)$ signifie « x aime y »)
- c^{f1} : fonction à un argument ($c(x)$ dénote le conjoint de x)

Une interprétation possible sur cette signature est l'interprétation I de domaine $D = \{0,1,2\}$ où :

- $Anne_I^{f0} = 0$, $Bernard_I^{f0} = 1$, et $Claude_I^{f0} = 2$
- $a_I^{r2} = \{(0,1),(1,0),(2,0)\}$
- c_I^{f1} est une fonction de D dans D qu'on définit par

x	0	1	2
$c_I^{f1}(x)$	1	0	2

Exemple 4.3.29

Nous obtenons :

- $[a(Anne, Bernard)]_I =$

  $\boxed{\textit{vrai} \text{ car } (\llbracket Anne \rrbracket_I, \llbracket Bernard \rrbracket_I) = (0,1) \in a_I^{r^2}.}$

- $[a(Anne, Claude)]_I =$

  $\boxed{\textit{faux} \text{ car } (\llbracket Anne \rrbracket_I, \llbracket Claude \rrbracket_I) = (0,2) \notin a_I^{r^2}.}$

Exemple 4.3.29

Soit e l'état $x = 0, y = 2$. Nous avons :

- $[a(x, c(x))]_{(I,e)} =$

  $\boxed{\textit{vrai} \text{ car } (\llbracket x \rrbracket_{(I,e)}, \llbracket c(x) \rrbracket_{(I,e)}) = (0, c_I^{f1}(0)) = (0, 1) \in a_I^{r2}.}$

- $[a(y, c(y))]_{(I,e)} =$

  $\boxed{\textit{faux} \text{ car } (\llbracket y \rrbracket_{(I,e)}, \llbracket c(y) \rrbracket_{(I,e)}) = (2, c_I^{f1}(2)) = (2, 2) \notin a_I^{r2}.}$

Ici on a utilisé *vrai* et *faux* au lieu des valeurs de vérité $0, 1$ pour les distinguer des éléments $0, 1$ du domaine (attention suivant le contexte).

Exemple 4.3.29

Nous avons :

- $[(Anne = Bernard)]_I =$

  $\boxed{faux, \text{ car } (\llbracket Anne \rrbracket_I, \llbracket Bernard \rrbracket_I) = (0,1) \quad et \quad (0,1) \notin =_I^{r2}.}$

- $[(c(Anne) = Anne)]_I =$

  $\boxed{faux, \text{ car } (\llbracket c(Anne) \rrbracket_I, \llbracket Anne \rrbracket_I) = (c_I^{f1}(0), 0) = (1,0).}$

- $[(c(c(Anne)) = Anne)]_I =$

  $\boxed{\begin{array}{l} vrai, \text{ car} \\ (\llbracket c(c(Anne)) \rrbracket_I, \llbracket Anne \rrbracket_I) = (c_I^{f1}(c_I^{f1}(0)), 0) = (0,0) \quad et \quad (0,0) \in =_I^{r2}. \end{array}}$

Sens des formules non atomiques 4.3.30

1. Les connecteurs propositionnels ont le même sens qu'en logique propositionnelle.
2. Notons $e[x = d]$ l'état identique à l'état e, sauf pour x.

$$[\forall x B]_{(I,e)} = min_{d \in D}[B]_{(I,e[x=d])} = \prod_{d \in D}[B]_{(I,e[x=d])},$$

vrai si $[B]_{(I,f)} = 1$ pour tout état f identique à e, sauf pour x.

3. $$[\exists x B]_{(I,e)} = max_{d \in D}[B]_{(I,e[x=d])} = \sum_{d \in D}[B]_{(I,e[x=d])},$$

vrai s'il y a un état f identique à e, sauf pour x, tel que $[B]_{(I,f)} = 1$.

Exemple 4.3.32

Utilisons l'interprétation I donnée dans l'exemple 4.3.19.
(Rappel $D = \{0,1,2\}$)

- $[\exists x\ a(x,x)]_I$

$$\boxed{= [a(0,0)]_I + [a(1,1)]_I + [a(2,2)]_I = \mathit{faux} + \mathit{faux} + \mathit{faux} = \mathit{faux}.}$$

- $[\forall x \exists y\ a(x,y)]_I$
$= \min\{[\exists y\ a(0,y)]_I,\ \ [\exists y\ a(1,y)]_I,\ \ [\exists y\ a(2,y)]_I\}$

$$\begin{aligned} = \min\{&\max\{[a(0,0)]_I, [a(0,1)]_I, [a(0,2)]_I\}, \\ &\max\{[a(1,0)]_I, [a(1,1)]_I, [a(1,2)]_I\}, \\ &\max\{[a(2,0)]_I, [a(2,1)]_I, [a(2,2)]_I\}\} \end{aligned}$$

$= \min\{\max\{\mathit{faux}, \mathit{vrai}, \mathit{faux}\}, \max\{\mathit{vrai}, \mathit{faux}, \mathit{faux}\},$
$\max\{\mathit{vrai}, \mathit{faux}, \mathit{faux}\}\}$

$= \min\{\mathit{vrai}, \mathit{vrai}, \mathit{vrai}\} = \mathit{vrai}.$

Exemple 4.3.32

- $[\exists y \forall x\ a(x,y)]_I$

$$
\begin{aligned}
&= [a(0,0)]_I.[a(1,0)]_I.[a(2,0)]_I + [a(0,1)]_I.[a(1,1)]_I.[a(2,1)]_I \\
&\quad + [a(0,2)]_I.\ [a(1,2)]_I.\ [a(2,2)]_I \\
&= \mathit{faux}.\mathit{vrai}.\mathit{vrai} + \mathit{vrai}.\mathit{faux}.\mathit{faux} + \mathit{faux}.\mathit{faux}.\mathit{faux} \\
&= \mathit{faux} + \mathit{faux} + \mathit{faux} = \mathit{faux}.
\end{aligned}
$$

Remarque 4.3.33

Les formules $\forall x \exists y\ a(x,y)$ et $\exists y \forall x\ a(x,y)$ n'ont pas la même valeur.
En intervertissant un $\exists$ et un $\forall$, on ne préserve pas le sens des formules.

Modèle, validité, conséquence, équivalence

Ces notions sont définies comme en logique propositionnelle mais...

Pour donner une valeur à une formule

- **En logique propositionnelle** : assignation $V \to \{0, 1\}$
- **En logique du premier ordre** : (I, e) où
  - I est une interprétation des symboles
  - e un état des variables.

... on utilise une interprétation au lieu d'une assignation.

La valeur d'une formule ne dépend que :

- de l'état de ses variables libres
- et de l'interprétation de ses symboles.

Plan

Sens des formules

Interprétation finie par expansion : compléments

Interprétation et substitution

Equivalences remarquables

Conclusion

Rappels à propos de la méthode des expansions

Recherche de modèles à n éléments **par réduction au cas propositionnel**

Cas simple : formule n'ayant ni symbole de fonction ni constante, sauf des entiers inférieurs à n.

Construction du modèle à n éléments

1. suppression des quantificateurs par expansion sur un domaine à n éléments,
2. remplacement des égalités par leur valeur
3. recherche d'une assignation propositionnelle des formules atomiques qui soit modèle de la formule.

Propriété de la n-expansion

Théorème 4.3.41

Soit A une formule ne comportant que des entiers $< n$.
Soit B la n-expansion de A.
Toute interprétation de domaine $\{0, \ldots, n-1\}$ attribue la même valeur à A et à B.

Preuve : par récurrence sur la taille des formules.

Assignation VS interprétation

Soit A une formule :

- fermée
- sans quantificateur,
- sans égalité ni symbole de fonction,
- sans constante sauf des entiers inférieurs à n.

Soit P l'ensemble des formules atomiques de A (sauf $\top$ et $\bot$).

Théorème 4.3.42

Pour toute assignation propositionnelle $v : P \to \{faux, vrai\}$
il existe une interprétation I de A telle que $[A]_I = [A]_v$.

Théorème 4.3.44

Pour toute interprétation I
il existe une assignation $v : P \to \{faux, vrai\}$ telle que $[A]_I = [A]_v$.

Exemple 4.3.43

Soit v l'assignation définie par $[p(0)]_v = \mathit{vrai}$ et $[p(1)]_v = \mathit{faux}$.

v donne la valeur *faux* à la formule $(p(0) + p(1)) \Rightarrow (p(0).p(1))$.

L'interprétation I définie par $p_I = \{0\}$ donne les mêmes valeurs aux mêmes formules.

v et I sont deux façons analogues de présenter une interprétation.

Correction de la méthode

$$\begin{array}{ccccccc} & n\text{-expansion} & & \text{simplifications} & & & \\ A & \longrightarrow & B & \longrightarrow & C & \longrightarrow & C \\ \text{(1er ordre)} & \equiv_n & \text{(sans } \forall\exists\text{)} & \equiv & \text{(sans const.)} & \approx & \text{(propos.)} \end{array}$$

- $[A]_I = [B]_I$ pour toute I sur un domaine de taille n
- $B \equiv C$ par construction (donc $[B]_I = [C]_I$ pour toute I)
-
  - À toute v correspond une I telle que $[C]_I = [C]_v$.
  - À toute I correspond une v telle que $[C]_I = [C]_v$.

A a donc un modèle I sur un domaine de taille n
si et seulement si
C a un modèle v (et v permet de retrouver I le cas échéant).

Recherche d'un modèle fini d'une formule fermée **avec** symbole de fonction

Soit A une formule fermée pouvant comporter des entiers de valeur inférieure à n.

Procédure

- Remplacer A par sa n-expansion.
- Enumérer les choix des valeurs des symboles, en propageant le plus possible chacun des choix effectués.

Similaire à *l'algorithme de* `DPLL`.

Exemple 4.3.46 : $A = \exists y P(y) \Rightarrow P(a)$

On cherche un contre-modèle à 2 éléments.

2-expansion de A

$$P(0) + P(1) \Rightarrow P(a)$$

Il reste à trouver les valeurs de $P(0)$, $P(1)$ et a.
On choisit (arbitrairement) $a = 0$:

$$P(0) + P(1) \Rightarrow P(0)$$

$P(0) = \mathit{faux}, P(1) = \mathit{vrai}$ est un contre-modèle propositionnel,
on en déduit une interprétation I telle que $P_I = \{1\}$.

Un contre-modèle est I de domaine $\{0, 1\}$ telle que $P_I = \{1\}$ et $a_I = 0$.

Exemple 4.3.47 : $P(a), \forall x(P(x) \Rightarrow P(f(x))), \neg P(f(b))$

1. 2-expansion :
   $F = \{P(a), (P(0) \Rightarrow P(f(0))).(P(1) \Rightarrow P(f(1))), \neg P(f(b))\}$.
2. On cherche des valeurs de $P(0)$, $P(1)$, a, b, $f(0)$ et $f(1)$ modèle de F.
3. Choisissons $a = 0$
   - De $P(a) = vrai$ et $a = 0$, on déduit : $P(0) = vrai$
   - De $P(0) = vrai$ et de $(P(0) \Rightarrow P(f(0))) = vrai$, on déduit : $P(f(0)) = vrai$
   - De $P(f(b)) = faux$ et de $P(f(0)) = vrai$, on déduit $f(0) \neq f(b)$ donc que $b \neq 0$, donc : $b = 1$ et $P(f(1)) = faux$
   - De $P(f(1)) = faux$ et $P(0) = vrai$, on déduit $f(1) \neq 0$ donc : $f(1) = 1$ et $P(1) = faux$
   - De $P(f(0)) = vrai$ et $P(1) = faux$, on déduit : $f(0) = 0$

Modèle : $a = 0, b = 1, P = \{0\}, f(0) = 0, f(1) = 1$

William McCune (1953-2011)

- Auteur de plusieurs systèmes de raisonnement automatisés : Otter, Prover9, Mace4

MACE

- **expansion** des formules du premier ordre
- **algorithmes performants** comme `DPLL`

http://www.cs.unm.edu/~mccune/mace4/examples/2009-11A/mace4-misc/

- 1996 : Preuve de la conjecture de Robbins à l'aide du logiciel de démonstration automatisée EQP
  - 8 jours de calcul sur un processeur à 66 MHz, 30 Mo de mémoire
  - production d'un témoin de preuve par Otter, vérifié par un troisième programme

(Conjecture sans réponse depuis 1933)

Plan

Sens des formules

Interprétation finie par expansion : compléments

Interprétation et substitution

Equivalences remarquables

Conclusion

Au niveau propositionnel

Substituer une variable propositionnelle dans une formule valide donne une formule valide.

Exemple :

Soit $\sigma(p) = \forall x\ q(x)$.
$p \vee \neg p$ est valide, il en est de même de la formule

$$\sigma(p \vee \neg p) = \forall x\ q(x) \vee \neg \forall x\ q(x)$$

Le principe de remplacement est encore valable aussi car :

Pour toutes formules A et B et toute variable x :

- $(A \Leftrightarrow B) \models (\forall x A \Leftrightarrow \forall x B)$
- $(A \Leftrightarrow B) \models (\exists x A \Leftrightarrow \exists x B)$

Instanciation d'une variable dans un terme

Définition 4.3.34

$A < x := t >$ est la formule obtenue en remplaçant dans A toute occurrence libre de x par t.

Exemple 4.3.35

Soit A la formule $(\forall x P(x) \vee Q(\mathbf{x}))$, la formule $A < x := b >$ vaut

$(\forall x P(x) \vee Q(b))$ car seule l'occurrence en gras de x est libre.

Mais on ne peut pas substituer n'importe quoi :

Exemple 4.3.37

Soit A la formule $\exists y p(x, y)$.

- $A < x := y > = \boxed{\exists y p(y, y)}$ (phénomène de capture)

La capture change le sens :

Exemple 4.3.37

Soit p une relation binaire interprétée sur $\{0,1\}$ par $p_I = \{(0,1)\}$
Soit e un état où $y = 0$.

- $[A < x := y >]_{(I,e)} =$

  $$\boxed{[\exists y p(y,y)]_{(I,e)} = [p(0,0)]_{(I,e)} + [p(1,1)]_{(I,e)} = faux + faux = faux.}$$

- Soit $d = 0$.
  Dans l'assignation $(I, e[x = d])$, nous avons $x = 0$.
  Donc $[A]_{(I,e[x=d])} =$

  $$\boxed{[\exists y p(x,y)]_{(I,e[x=d])} = [p(0,0)]_{(I,e)} + [p(0,1)]_{(I,e)} = faux + vrai = vrai.}$$

Ainsi, $[A < x := y >]_{(I,e)} \neq [A]_{(I,e[x=d])}$, pour $d = [\![y]\!]_{(I,e)}$.

Instanciation d'une variable dans un terme : précautions

Solution : notion de terme t libre pour une variable

Définition 4.3.34

2. t est libre pour x dans A si les variables de t ne sont pas liées dans les occurrences libres de x.

Exemple 4.3.35

- Le terme $f(z)$ est libre pour x dans la formule $\exists y p(x, y)$.
- Par contre les termes y ou $g(y)$ ne sont pas libres pour x dans cette formule.
- Par définition, le terme x est libre pour x dans toute formule.

Propriétés

Théorème 4.3.36

Soient A une formule et t un terme libre pour la variable x dans A.
Pour toute assignation (I, e) nous avons
$[A < x := t >]_{(I,e)} = [A]_{(I,e[x=d])}$ où $d = [\![t]\!]_{(I,e)}$.

Corollaire 4.3.38

Soient A une formule et t un terme libre pour x dans A.
Les formules $\forall x A \Rightarrow A < x := t >$ et $A < x := t > \Rightarrow \exists x A$ sont valides.

Plan

Sens des formules

Interprétation finie par expansion : compléments

Interprétation et substitution

Equivalences remarquables

Conclusion

Relations entre $\forall$ et $\exists$

Lemme 4.4.1

Soient A une formule et x une variable :

$$\begin{aligned} \neg\forall x A &\equiv \exists x \neg A \\ \forall x A &\equiv \neg\exists x \neg A \\ \neg\exists x A &\equiv \forall x \neg A \\ \exists x A &\equiv \neg\forall x \neg A \end{aligned}$$

On prouve les deux premières identités.

Preuve de $\neg\forall x A \equiv \exists x \neg A$

$$
\begin{aligned}
[\neg\forall x A]_{(I,e)} &= 1 - [\forall x A]_{(I,e)} \\
&= 1 - \min_{d\in D}[A]_{(I,e[x=d])} \\
&= \max_{d\in D}\left(1 - [A]_{(I,e[x=d])}\right) \\
&= \max_{d\in D}[\neg A]_{(I,e[x=d])} \\
&= [\exists x \neg A]_{(I,e)}
\end{aligned}
$$

Preuve de $\forall x A \equiv \neg\exists x \neg A$:

Transformons $\forall x A$

$\equiv \neg\neg\forall x A$

$\equiv \neg\exists x \neg A$ (ci-dessus)

Déplacement des quantificateurs

Soient x, y deux variables et A, B deux formules.

1. $\forall x \forall y A \equiv \forall y \forall x A$
2. $\exists x \exists y A \equiv \exists y \exists x A$
3. $\forall x (A \wedge B) \equiv (\forall x A \wedge \forall x B)$
4. $\exists x (A \vee B) \equiv (\exists x A \vee \exists x B)$
5. Soient Q un quantificateur et $\circ$ un des connecteurs $\wedge, \vee$
   Si x n'est pas une variable libre de A alors :
   5.1 $Qx\ A \equiv A$,
   5.2 $Qx\ (A \circ B) \equiv A \circ Qx\ B$

Exemple 4.4.2

Nous éliminons de ces deux formules les quantificateurs inutiles :

- $\forall x \exists x P(x) \equiv$

  $\boxed{\exists x P(x)}$

- $\forall x (\exists x P(x) \vee Q(x)) \equiv$

  $\boxed{\exists x P(x) \vee \forall x Q(x)}$

Changement de variables liées (1/3)

Théorème 4.4.3

Soit Q un quantificateur.
Si y ne figure pas dans $Qx\ A$ alors : $Qx\ A \equiv Qy\ A < x := y >$

Exemple 4.4.4

- $\forall x\ p(x,z) \equiv \forall y\ p(y,z)$
- $\forall x\ p(x,z) \not\equiv \forall z\ p(z,z)$

Plan

Sens des formules

Interprétation finie par expansion : compléments

Interprétation et substitution

Equivalences remarquables

Conclusion

Aujourd'hui

- Évaluer une formule = choisir une interprétation pour ses **symboles** et un état pour ses **variables**
- Adaptation de la méthode des **expansions** aux formules avec symboles de fonction
- Équivalences remarquables à propos des quantificateurs (attention, **pas de notion de forme normale** utilisable)

La prochaine fois

- Skolémisation
- Semi-algorithme pour montrer qu'une formule est insatisfaisable.

À chercher

Tous les hommes sont mortels.
Socrate est un homme.
Donc Socrate est mortel.

- Rechercher un contre-modèle par 1-expansion puis par 2-expansion.
- Que peut-on en conclure ?

Base de la démontration automatique : Théorème de Herbrand et Skolémisation

Benjamin Wack

Université Grenoble Alpes

Mars 2022

Rappel sur l'expansion

Tous les hommes sont mortels.	$\forall x(homme(x) \Rightarrow mortel(x))$
Socrate est un homme.	$\wedge homme(Socrate)$
Donc Socrate est mortel.	$\Rightarrow mortel(Socrate)$

- Rechercher un contre-modèle par 1-expansion puis par 2-exp.
  - 1-expansion :
    $(homme(0) \Rightarrow mortel(0)).homme(Socrate) \Rightarrow mortel(Socrate)$
    On est forcé d'interpréter *Socrate* comme 0 : pas de c.-m.
  - 2-expansion : $(homme(0) \Rightarrow mortel(0)).$
    $(homme(1) \Rightarrow mortel(1)).homme(Socrate) \Rightarrow mortel(Socrate)$
    On peut interpréter *Socrate* comme 0 ou 1, mais aucun ne donne de contre-modèle.
- Que peut-on en conclure ?
  Rien ! Si ce n'est que cette formule est satisfaisable.

Plan

Introduction

Domaine et base de Herbrand

Interprétation de Herbrand

Théorème de Herbrand

Skolémisation
- Motivation, propriétés, exemples
- Définitions et procédure

Conclusion

Plan

Introduction

Domaine et base de Herbrand

Interprétation de Herbrand

Théorème de Herbrand

Skolémisation
- Motivation, propriétés, exemples
- Définitions et procédure

Conclusion

Introduction

En logique du premier ordre, il n'y a pas d'algorithme pour décider si une formule est valide ou non valide.

Algorithme de semi-décision :

1. S'il termine alors il décide correctement si la formule est valide. Lorsque la formule est valide, la décision est généralement accompagnée d'une preuve.
2. Si la formule est valide, alors il termine. Cependant, l'exécution peut être longue !

Si la formule n'est pas valide, la terminaison d'un tel algorithme n'est pas garantie.

Plan

Introduction

Domaine et base de Herbrand

Interprétation de Herbrand

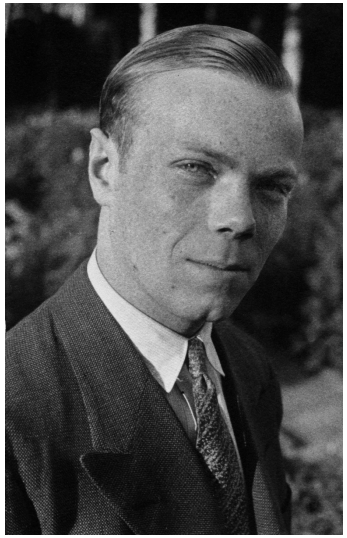
Théorème de Herbrand

Skolémisation
- Motivation, propriétés, exemples
- Définitions et procédure

Conclusion

Jacques Herbrand (1908-1931)

- Résultats en théorie des nombres
- 1930 : réduction de la validité d'une formule du premier ordre à un ensemble de formules propositionnelles
- Correspondance avec Gödel à propos de la cohérence de l'arithmétique



Fermeture universelle

Définition 5.1.1

Soit C une formule ayant pour variables libres $x_1, \ldots, x_n$.

La fermeture universelle de C, notée $\forall(C)$, est la formule $\forall x_1 \ldots \forall x_n C$.

Exemple 5.1.2

$\forall(\ P(x) \wedge R(x,y)\ ) =$

$$\boxed{\forall x \forall y (P(x) \wedge R(x,y)) \quad \text{ou} \quad \forall y \forall x (P(x) \wedge R(x,y))}$$

Soit Γ un ensemble de formules : $\forall(\Gamma) = \{\ \forall(A) \mid A \in \Gamma\ \}$.
Par exemple : $\forall(\ \{P(x), Q(x)\}\ ) = \{\forall x P(x), \forall x Q(x)\}$

Hypothèses

Nous ne considérons que :

- des formules sans $=$, $\top$ ni $\bot$ (car leur sens est fixé).
- des signatures qui comportent au moins une constante (quitte à ajouter **la constante *a***).

Domaine et base de Herbrand

Définition 5.1.4

1. Domaine de Herbrand $\boldsymbol{D_\Sigma}$ = ensemble des termes fermés de Σ (*i.e.* sans variable)
   Remarque : il n'est jamais vide, car $a \in D_\Sigma$.
2. Base de Herbrand $\boldsymbol{B_\Sigma}$ = ensemble des formules atomiques fermées de Σ

Exemple 5.1.5

1. Soit $\Sigma = \{a^{f0}, b^{f0}, P^{r1}, Q^{r1}\}$: $D_\Sigma = \{a, b\}$ et
   $$\boxed{B_\Sigma = \{P(a), P(b), Q(a), Q(b)\}.}$$
2. Soit $\Sigma = \{a^{f0}, f^{f1}, P^{r1}\}$: $D_\Sigma = \{f^n(a) \mid n \in \mathbb{N}\}$ et
   $$\boxed{B_\Sigma = \{P(f^n(a)) \mid n \in \mathbb{N}\}}$$

Plan

Introduction

Domaine et base de Herbrand

Interprétation de Herbrand

Théorème de Herbrand

Skolémisation
- Motivation, propriétés, exemples
- Définitions et procédure

Conclusion

Interprétation de Herbrand

Définition 5.1.6

Soit $E \subseteq B_\Sigma$.
L'interprétation de Herbrand $H_{\Sigma,E}$ a pour domaine D_Σ et :

1. Si s est une constante, il est interprété par lui-même.
2. Si s est un symbole de fonction et si $t_1, \ldots, t_n \in D_\Sigma$ alors $s_{H_{\Sigma,E}}(t_1, \ldots, t_n) = s(t_1, \ldots, t_n)$.
3. Si s est une variable propositionnelle, $s_{H_{\Sigma,E}} = 1$ ssi $s \in E$.
4. Si s est un symbole de relation alors $s_{H_{\Sigma,E}} = \{(t_1, \ldots, t_n)$ tels que $s(t_1, \ldots, t_n) \in E\}$.

Autrement dit :
- Les termes sont interprétés par eux-mêmes.
- E est l'ensemble des formules atomiques vraies.

Exemple 5.1.8

Soit $\Sigma = \{a^{f0}, b^{f0}, P^{r1}, Q^{r1}\}$

Le domaine de Herbrand est $D_\Sigma = \{a, b\}$.

$E = \{P(b), Q(a)\}$ définit l'interprétation de Herbrand H où :

- les constantes a et b ont pour valeur elles-mêmes
- $P_H = \{b\}$ et $Q_H = \{a\}$

Formule universelle et modèle de Herbrand

Théorème 5.1.16

Soit Γ un ensemble de formules sans quantificateur sur la signature Σ.

$$\forall(\Gamma) \text{ a un modèle}$$

si et seulement si

$\forall(\Gamma)$ a un modèle qui est une interprétation de Herbrand.

- Preuve : cf poly (il « suffit » de bien choisir E)
- Conséquence : pas la peine d'en chercher une autre !

Exemple

Soit $\Sigma = \{a^{f0}, b^{f0}, P^{r1}, Q^{r1}\}$

Soit I l'interprétation de domaine $\{0, 1\}$ où :

- $a_I = 0$, $b_I = 1$,
- $P_I = \{1\}$ et $Q_I = \{0\}$.

Le domaine de Herbrand est $D_\Sigma = \{a, b\}$.

$E = \{P(b), Q(a)\}$ définit l'interprétation de Herbrand H où :

- les constantes a et b ont pour valeur elles-mêmes
- $P_H = \{b\}$ et $Q_H = \{a\}$

I est modèle d'un ensemble $\forall(\Gamma)$ ssi H est un modèle de $\forall(\Gamma)$.

Plan

Introduction

Domaine et base de Herbrand

Interprétation de Herbrand

Théorème de Herbrand

Skolémisation
- Motivation, propriétés, exemples
- Définitions et procédure

Conclusion

Théorème de Herbrand

Théorème 5.1.17

Soit Γ un ensemble de formules sans quantificateur de signature Σ.

$\forall(\Gamma)$ a un modèle

si et seulement si

Tout ensemble fini d'instances fermées des formules de Γ a un modèle propositionnel $B_\Sigma \to \{0,1\}$.

Rappels :

- Σ comporte au moins une constante a et pas de signe $=$
- Instancier = substituer un terme à chaque variable

Variante du théorème de Herbrand

Corollaire 5.1.18

Soit Γ un ensemble de formules sans quantificateur de signature Σ.

$\forall(\Gamma)$ est insatisfaisable

si et seulement si

Il existe un ensemble fini insatisfaisable d'instances fermées des formules de Γ.

Preuve.

C'est la « contraposée » du théorème de Herbrand. □

Procédure de semi-décision : insatisfaisabilité de $\forall(\Gamma)$

Soit Γ un ensemble fini de formules sans quantificateur.
Énumérer l'ensemble des instances fermées des formules de Γ et :

1. si on trouve un ensemble insatisfaisable, $\forall(\Gamma)$ est insatisfaisable.
2. si on termine sans contradiction (pour un Σ *sans fonction*), $\forall(\Gamma)$ est satisfaisable.
3. en attendant, on ne peut pas conclure :
   - soit $\forall(\Gamma)$ est satisfaisable (et on ne s'arrêtera jamais) ;
   - soit $\forall(\Gamma)$ est insatisfaisable, et on n'a pas encore énuméré assez d'instances pour obtenir une contradiction.

Exemple 5.1.19 (1/5)

Soit $\Gamma = \{P(x), Q(x), \neg P(a) \vee \neg Q(b)\}$ et $\Sigma = \{a^{f0}, b^{f0}, P^{r1}, Q^{r1}\}$.

$D_\Sigma = \{a, b\}$.

L'ensemble $\{P(a), Q(b), \neg P(a) \vee \neg Q(b)\}$ d'instances sur D_Σ est insatisfaisable, donc $\forall(\Gamma)$ est insatisfaisable.

Exemple 5.1.19 (2/5)

Soit $\Gamma = \{P(x) \vee Q(x), \neg P(a), \neg Q(b)\}$

L'ensemble de toutes les instances sur D_Σ est :
$\{P(a) \vee Q(a),\ \ P(b) \vee Q(b),\ \ \neg P(a),\ \ \neg Q(b)\}$
Il a un modèle propositionnel caractérisé par $E = \{P(b), Q(a)\}$.

Donc l'interprétation de Herbrand $H_{\Sigma,E}$ est modèle de $\forall(\Gamma)$.

Exemple 5.1.19 (3/5)

Soit $\Gamma = \{P(x), \neg P(f(x))\}$ et $\Sigma = \{a^{f0}, f^{f1}, P^{r1}\}$.

$D_\Sigma = \{f^n(a) \mid n \in \mathbb{N}\}$.

L'ensemble $\{P(f(a)), \neg P(f(a))\}$ est insatisfaisable,
donc $\forall(\Gamma)$ est insatisfaisable.

Exemple 5.1.19 (4/5)

$$\text{Soit } \Gamma = \left\{ \begin{array}{l} \neg P(a), \\ P(x) \vee \neg P(f(x)), \\ P(f(f(a))) \end{array} \right\}$$

$$\boxed{\left\{ \begin{array}{l} \neg P(a), \\ P(a) \vee \neg P(f(a)), \\ P(f(a)) \vee \neg P(f(f(a))), \\ P(f(f(a))) \end{array} \right\} \text{ est insatisfaisable, donc } \forall(\Gamma) \text{ aussi.}}$$

Remarque : observez qu'il a fallu prendre 2 instances ($x := a$ puis $x := f(a)$) de la deuxième formule de Γ pour obtenir une contradiction.

Exemple 5.1.19 (5/5)

$$\text{Soit } \Gamma = \left\{ \begin{array}{l} R(x, s(x)), \\ R(x,y) \wedge R(y,z) \Rightarrow R(x,z), \\ \neg R(x,x) \end{array} \right\} \quad \begin{array}{l} n < n+1 \\ x < y < z \Rightarrow x < z \\ \neg(x < x) \end{array}$$

et $\Sigma = \{a^{f0}, s^{f1}, R^{r2}\}$.

$D_\Sigma = \{s^n(a) \mid n \in \mathbb{N}\}$ (domaine infini)

Tout ensemble fini d'instances des formules de Γ a un modèle : leur énumération ne s'arrêtera jamais.

En effet, $\forall(\Gamma)$ a un modèle (infini) : l'interprétation I de domaine $\mathbb{N}$ avec $a_I = 0$, $\quad s_I(n) = n+1$ et $R_I(x,y) = x < y$.

Remarque : $\forall(\Gamma)$ n'a aucun modèle fini (inutile d'en chercher un par n-expansion).

Plan

Introduction

Domaine et base de Herbrand

Interprétation de Herbrand

Théorème de Herbrand

Skolémisation
- Motivation, propriétés, exemples
- Définitions et procédure

Conclusion

Pourquoi la Skolémisation ?

Le théorème de Herbrand s'applique à la fermeture universelle d'un ensemble de formules sans quantificateur.

Pour des formules avec quantificateur existentiel on utilise la skolémisation (Thoralf Albert Skolem).

La skolémisation

- change un ensemble de formules fermées en la fermeture universelle d'un ensemble de formules sans quantificateur.
- préserve l'existence d'un modèle (la satisfaisabilité).

Exemple 5.2.1

La formule $\exists x P(x)$ est skolémisée en $P(a)$.

On observe les relations suivantes entre ces deux formules :

1. $P(a)$ a pour conséquence $\exists x P(x)$
2. $\exists x P(x)$ n'a pas pour conséquence $P(a)$ mais un modèle de $\exists x\, P(x)$ « donne » un modèle de $P(a)$.

   (Il suffit d'interpréter a comme un élément de P_I.)

Définitions

Une formule est en forme normale si elle n'a ni $\Leftrightarrow$ ni $\Rightarrow$ et les négations portent uniquement sur les **formules atomiques**.

Définition 5.2.3

Une formule fermée est propre si aucune variable n'est liée par deux quantificateurs distincts.

Exemple 5.2.4

- La formule $\forall x P(x) \vee \forall x Q(x)$ n'est **pas propre**.
- La formule $\forall x P(x) \vee \forall y Q(y)$ est **propre**.
- La formule $\forall x (P(x) \Rightarrow \exists x Q(x) \wedge \exists y R(x, y))$ n'est **pas propre**.
- La formule $\forall x (P(x) \Rightarrow \exists y R(x, y))$ est **propre**.

Comment skolémiser une formule fermée A ?

Définition 5.2.5 (skolémisation)

Soit A une formule fermée :

1. B = Normaliser A.
2. C = Rendre B propre.
3. D = Éliminer les quantificateurs existentiels de C.
   Cette transformation préserve seulement l'existence de modèle.
4. E = Enlever les quantificateurs universels de D.

E est la forme de Skolem de A.
(E est une formule normale sans quantificateur.)

1. Normalisation

1. Remplacer les équivalences
2. Remplacer les implications
3. Déplacer les négations vers les formules atomiques

Règles

1. et 2. Comme dans le cas propositionnel : $\begin{cases} A \Leftrightarrow B \equiv (A \Rightarrow B) \wedge (B \Rightarrow A) \\ A \Rightarrow B \equiv \neg A \vee B \end{cases}$

3. Comme dans le cas propositionnel : $\begin{cases} \neg\neg A \equiv A \\ \neg(A \wedge B) \equiv \neg A \vee \neg B \\ \neg(A \vee B) \equiv \neg A \wedge \neg B \end{cases}$

De plus $\begin{cases} \neg\forall x A \equiv \exists x \neg A \\ \neg\exists x A \equiv \forall x \neg A \end{cases}$

Exemple 5.2.7

La forme normale de $\forall y(\forall x P(x,y) \Leftrightarrow Q(y))$ est :

On remplace $\Leftrightarrow$:

$$\forall y((\neg \forall x P(x,y) \vee Q(y)) \wedge (\neg Q(y) \vee \forall x P(x,y)))$$

puis on déplace $\neg$:

$$\forall y((\exists x \neg P(x,y) \vee Q(y)) \wedge (\neg Q(y) \vee \forall x P(x,y)))$$

2. Transformation en formule propre

Renommer les variables liées, par exemple en choisissant de nouveaux noms.

Exemple 5.2.8

- La formule $\forall x P(x) \vee \forall x Q(x)$ est changée en

$$\boxed{\forall x P(x) \vee \forall y Q(y)}$$

- La formule $\forall x (P(x) \Rightarrow \exists x Q(x) \wedge \exists y R(x, y))$ est changée en

$$\boxed{\forall x (P(x) \Rightarrow \exists z Q(z) \wedge \exists y R(x, y))}$$

Rappel : renommage des variables liées

Théorème 4.4.3

Soit Q un quantificateur.
Si y ne figure pas dans $Qx\ A$ alors : $Qx\ A \equiv Qy\ A < x := y >$

Exemple 4.4.4

- $\forall x\ p(x,z) \equiv \forall y\ p(y,z)$
- $\forall x\ p(x,z) \not\equiv \forall z\ p(z,z)$

3. Élimination des quantificateurs existentiels

Soit $\exists y B$ une sous-formule d'une formule A fermée normale et propre.
Soient $x_1, \ldots x_n$ l'ensemble des variables libres de $\exists y B$.

On prend f un nouveau symbole (si $n = 0$, f est une constante)
et on remplace $\exists y B$ par $B < y := f(x_1, \ldots x_n) >$.

Théorème 5.2.9

La formule A' obtenue est fermée, normale, propre et vérifie :

1. A' a pour conséquence A
2. Si A a un modèle alors A' a un modèle identique (sauf pour f).

Remarque 5.2.10

La formule obtenue reste fermée, normale et propre.

Par application répétée en choisissant un **nouveau** symbole à chaque quantificateur éliminé, on obtient :

- une formule B fermée, normale, propre et sans $\exists$
- telle que A a un modèle si et seulement si B en a un.

Exemple 5.2.11

En éliminant les quantificateurs existentiels de la formule $\exists x \forall y P(x,y) \wedge \exists z \forall u \neg P(z,u)$ on obtient

$\forall y P(a,y) \wedge \forall u \neg P(b,u)$
Il est facile d'en trouver un modèle.

Mais Si on fait l'erreur d'éliminer les deux $\exists$ avec la même constante a, on obtient $\forall y P(a,y) \wedge \forall u \neg P(a,u)$

qui est insatisfaisable (elle a pour conséquence $P(a,a) \wedge \neg P(a,a)$).

Exemple 5.2.12

En éliminant les quantificateurs existentiels de la formule $\exists x \forall y \exists z P(x, y, z)$ nous obtenons

deux solutions possibles :

- si on élimine $\exists x$ d'abord :
  $\forall y \exists z P(a, y, z) \quad \rightarrow \quad \forall y P(a, y, f(y))$
- si on élimine $\exists z$ d'abord :
  $\exists x \forall y P(x, y, g(x, y)) \quad \rightarrow \quad \forall y P(b, y, g(b, y))$

L'existence d'un modèle est préservée dans les deux cas.

4. Transformation en formule universelle

Théorème 5.2.13

Soit A une formule fermée, normale, propre et sans quantificateur existentiel.
Soit B la formule obtenue en enlevant tous les $\forall$.

A est équivalente à $\forall(B)$.

Démonstration.

Cela revient à effectuer tous les remplacements de la forme

- $(\forall x C) \wedge D \equiv \forall x(C \wedge D)$
- $(\forall x C) \vee D \equiv \forall x(C \vee D)$

où x est non libre dans D $\square$

Propriété de la skolémisation

Propriété 5.2.14

Soit A une formule fermée et E sa forme de Skolem :
A a un modèle si et seulement si $\forall(E)$ a un modèle.

Démonstration.

A une formule fermée

$\downarrow$ Normaliser *(équivalente)*

B

$\downarrow$ Rendre propre *(équivalente)*

C

$\downarrow$ Élim. $\exists$ *("préserve" les modèles)*

D

$\downarrow$ Enlever $\forall$ *(équivalente à $\forall(E)$)*

E forme de Skolem

□

Exemple 5.2.15

Soit $A = \forall x(P(x) \Rightarrow Q(x)) \Rightarrow (\forall x P(x) \Rightarrow \forall x Q(x))$. On skolémise $\neg A$.

1. $\neg A$ est transformée en la formule normale :
   $\forall x(\neg P(x) \vee Q(x)) \wedge \forall x P(x) \wedge \exists x \neg Q(x)$
2. La formule normale est transformée en la formule propre :
   $\forall x(\neg P(x) \vee Q(x)) \wedge \forall y P(y) \wedge \exists z \neg Q(z)$
3. Le quantificateur existentiel est « remplacé » par une constante :
   $\forall x(\neg P(x) \vee Q(x)) \wedge \forall y P(y) \wedge \neg Q(a)$
4. Les quantificateurs universels sont enlevés :
   $(\neg P(x) \vee Q(x)) \wedge P(y) \wedge \neg Q(a)$.

L'instanciation $x := a, y := a$ donne $(\neg P(a) \vee Q(a)) \wedge P(a) \wedge \neg Q(a)$.

Donc (théorème de Herbrand) la forme de Skolem de $\neg A$ est insatisfaisable.

Puisque la skolémisation préserve la satisfaisabilité, $\neg A$ est insatisfaisable.

Plan

Introduction

Domaine et base de Herbrand

Interprétation de Herbrand

Théorème de Herbrand

Skolémisation
- Motivation, propriétés, exemples
- Définitions et procédure

Conclusion

Aujourd'hui

- Pour montrer que A est satisfaisable :
  - Recherche d'un modèle (fini) par n**-expansions**
- Pour montrer que A est insatisfaisable :
  - **Skolemisation**
  - Recherche d'un **ensemble (fini) insatisfaisable d'instances** sur D_Σ
  - Théorème de Herbrand : alors A l'est aussi
- Méthodes non terminantes et limitées aux interprétations finies
- Pour rechercher un contre-modèle ou prouver la validité de A, on procède de même avec $\neg A$

La prochaine fois

Méthode de déduction au premier ordre :

- Forme clausale
- Unification
- Résolution au premier ordre
- Cohérence
- Complétude

Base de la démontration automatique : Résolution au premier ordre

Benjamin Wack

Université Grenoble Alpes

Avril 2023

Plan

Introduction

Forme clausale

Unification

Résolution au 1er ordre

Complétude

Conclusion

Plan

Introduction

Forme clausale

Unification

Résolution au 1er ordre

Complétude

Conclusion

Idée

La skolémisation donne des formules sans quantificateur.

On en cherche ensuite des instances insatisfaisables (intuitivement ou par énumération).

Aujourd'hui, généralisation au premier ordre de la résolution :

- Mise en forme clausale des formes de Skolem
- Résolution sur des clauses avec variables
- Cohérence et complétude de la méthode

Plan

Introduction

Forme clausale

Unification

Résolution au 1er ordre

Complétude

Conclusion

Littéral, clause

Définition 5.2.19

Un littéral positif est une formule atomique. Ex : $P(x, y)$

Un littéral négatif est la négation d'une formule atomique. Ex : $\neg Q(a)$

Une clause est une somme de littéraux. Ex : $P(x, y) \vee \neg Q(a)$

Forme clausale d'une formule

Définition 5.2.20

La forme clausale d'une formule fermée A est obtenue en 2 étapes :

1. skolémiser A (on obtient une forme normale sans quantificateur)
2. distribuer les $\vee$ sur les $\wedge$ pour obtenir un ensemble de clauses Γ

Propriété 5.2.21

$\forall(\Gamma)$ a un modèle si et seulement A a un modèle. Plus précisément :

- $\forall(\Gamma)$ a pour conséquence A
- si A a un modèle alors $\forall(\Gamma)$ en a un aussi

Preuve : On le sait déjà pour la forme de Skolem.
Or cette dernière est équivalente à la forme clausale (distributivité).

Forme clausale d'un ensemble de formules

Définition 5.2.22

Soit $\Gamma = A_1, \ldots, A_n$ un ensemble de formules fermées.
La forme clausale de Γ est l'union des formes clausales de $A_1, \ldots, A_n$, en prenant soin d'éliminer chaque $\exists$ à l'aide d'un nouveau symbole.

Corollaire 5.2.23

Soient Γ un ensemble de formules fermées et Δ sa forme clausale :

- $\forall(\Delta)$ a pour conséquence Γ
- si Γ a un modèle alors $\forall(\Delta)$ a un modèle.

Adaptation du théorème de Herbrand aux formes clausales

Théorème 5.2.24

Soient Γ un ensemble de formules fermées et Δ sa forme clausale :

Γ est insatisfaisable

si et seulement si

il existe un ensemble fini insatisfaisable d'instances des clauses de Δ.

Preuve.

- La skolémisation préserve la satisfaisablité.
- Puis application du théorème de Herbrand à $\forall(\Delta)$.

$\square$

Exemple 5.2.25

Soit $A = \exists y \forall z (P(z,y) \Leftrightarrow \neg \exists x (P(z,x) \wedge P(x,z)))$. Calculons la forme clausale de A.

1-4. Les 4 étapes de Skolémisation donnent la forme :
$(\neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)) \ \wedge \ (P(z,f(z)) \wedge P(f(z),z) \ \vee \ P(z,a))$

5. Forme clausale :
   - $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
   - $C_2 = P(z,f(z)) \vee P(z,a)$
   - $C_3 = P(f(z),z) \vee P(z,a)$

On recherche un ensemble fini insatisfaisable d'instances de C_1, C_2, C_3.
On instancie :

- C_1 avec $x := a, z := a$ d'où : $C_1' = \neg P(a,a)$
- C_2 avec $z := a$ d'où : $C_2' = P(a,f(a)) \vee P(a,a)$
- C_3 avec $z := a$ d'où : $C_3' = P(f(a),a) \vee P(a,a)$
- C_1 avec $x := a, z := f(a)$ d'où : $C_1'' = \neg P(f(a),a) \vee \neg P(a,f(a))$

L'ensemble C_1', C_1'', C_2', C_3' est insatisfaisable, donc A est insatisfaisable.

Comment procéder en pratique

Soit Γ un ensemble de clauses. On veut montrer que $\forall(\Gamma)$ n'a pas de modèle :

- Comment choisir judicieusement les instances ?
- Comment démontrer leur insatisfaisabilité ?

Le système de « factorisation, copie, résolution binaire » est un système formel permettant de déduire $\perp$ de Γ.

Sa complétude est basée sur le théorème de Herbrand.
Pour trouver les instances contradictoires des clauses, les règles utilisent l'unification.

Plan

Introduction

Forme clausale

Unification

Résolution au 1er ordre

Complétude

Conclusion

John Alan Robinson (1930-2016)

- à l'origine du **principe de résolution**
- algorithme d'**unification** (1965)
  - rend efficace la recherche d'instances de clauses contradictoires
  - cas particulier du *filtrage* utilisé dans les langages fonctionnels
- Fondateur de la *programmation logique*

```
parent(pascal, mathilde).
frere(stephane, pascal).
oncle(X,Y) :- parent(Z,Y),
              frere(X,Z).
?- oncle(stephane, mathilde).
  true.
```

(Prolog, Colmerauer & Roussel, 1972)

Unification : expression, solution

Définition 5.3.1

- Un terme ou un littéral est une **expression**.
- Une substitution σ est **solution** de l'**équation** $e_1 = e_2$ si $e_1\sigma$ et $e_2\sigma$ sont syntaxiquement identiques.
- Une substitution est solution d'un ensemble d'équations si elle est solution de chaque équation de l'ensemble.

Unification : exemple 5.3.4

L'équation $P(x, f(y)) = P(g(z), z)$ a pour solution :

$$\boxed{x := g(f(y)), z := f(y)}$$

Le système d'équations $x = g(z), \ f(y) = z$ a pour solution :

$$\boxed{x := g(f(y)), z := f(y)}$$

Unification : composition de substitutions

Définition 5.3.5

- Soient σ et τ deux substitutions, on note $\sigma\tau$ la substitution définie par $x\sigma\tau = (x\sigma)\tau$ pour toute variable.
- La substitution $\sigma\tau$ est une instance de σ.
- Deux substitutions sont équivalentes si chacune d'elles est une instance de l'autre.

Unification : exemple 5.3.6

Considérons les substitutions

- $\sigma_1 =< x := g(z), y := z >$
- $\sigma_2 =< x := g(y), z := y >$
- $\sigma_3 =< x := g(a), y := a, z := a >$

On a les relations suivantes entre ces susbstitutions :

- $\sigma_1 = \sigma_2 < y := z >$
- $\sigma_2 = \sigma_1 < z := y >$

σ_1 et σ_2 sont équivalentes.

- $\sigma_3 = \sigma_1 < z := a >$
- $\sigma_3 = \sigma_2 < y := a >$

σ_3 est une instance de σ_1 ou de σ_2, mais ne leur est pas équivalente.

Unification : définition de la solution la plus générale

Définition 5.3.7 (mgu)

Une solution d'un système d'équations est la plus générale si toute autre solution en est une instance.
Remarque : deux solutions « les plus générales » sont équivalentes.

Exemple 5.3.8

Considérons l'équation $f(x, g(z)) = f(g(y), x)$.

- $\sigma_1 =< x := g(z), y := z >$
- $\sigma_2 =< x := g(y), z := y >$
- $\sigma_3 =< x := g(a), y := a, z := a >$

sont 3 solutions.

σ_1 et σ_2 sont les plus générales.

Unificateur

Définition 5.3.2

Soit E un ensemble d'expressions et $E\sigma = \{t\sigma \mid t \in E\}$.
σ est un unificateur de E si et seulement si $E\sigma$ est réduit à un élément.

Si $E = \{e_1, \ldots, e_n\}$ cela revient à dire que

σ est solution du système d'équations $\begin{cases} e_1 = e_2 \\ \ldots \\ e_{n-1} = e_n \end{cases}$

On définit de même un unificateur le plus général (ou principal).

Unification : l'algorithme (plan)

L'algorithme distingue :

- les équations à résoudre, notées par un $=$
- les équations résolues, notées par un $:=$

Initialement, il n'y a pas d'équations résolues.

L'algorithme procède par systèmes équivalents et s'arrête quand :

- il n'y a plus d'équations à résoudre,
  alors les équations résolues donnent la solution la plus générale ;
- ou il a prouvé que le système n'a pas de solution.

Unification : l'algorithme (les règles)

Choisir une équation à résoudre puis :

1. **Supprimer l'équation** si ses 2 membres sont identiques.
2. **Décomposer**
   - $\neg A = \neg B$ devient $A = B$
   - $f(s_1, \ldots, s_n) = f(t_1, \ldots, t_n)$ devient $s_1 = t_1, \ \ldots, \ s_n = t_n$
     (rien si f est une constante)
3. **Échec de décomposition**
   Si l'équation est $f(s_1, \ldots, s_n) = g(t_1, \ldots, t_p)$ avec $f \neq g$
   alors il n'y a pas de solution.
   (en particulier si l'équation est $\neg A = B$ avec B un littéral positif)

Unification : l'algorithme (les règles)

4. **Orienter**
   Si l'équation est $t = x$ avec t un (vrai) terme et x une variable, on la remplace par $x = t$.
5. **Élimination d'une variable**
   Si l'équation est $x = t$ avec x une variable et t un terme ne contenant pas x :
   - l'enlever des équations à résoudre
   - remplacer x par t partout (équations à résoudre et résolues)
   - ajouter $x := t$ aux équations résolues
6. **Échec de l'élimination**
   Si l'équation est $x = t$ avec x une variable et t contenant x alors il n'y a pas de solution.

Unification : l'algorithme (exemple 5.3.11)

1. Résoudre $f(x, g(z)) = f(g(y), x)$.

Décomposition	$x = g(y), g(z) = x$
Élimination de x	$x := g(y), g(z) = g(y)$
Décomposition	$x := g(y), z = y$
Élimination de z	$x := g(y), z := y$ solution

2. Résoudre $f(x, x, a) = f(g(y), g(a), y)$.

Décomposition	$x = g(y), x = g(a), a = y$
Élimination de x dans la 1re équation	$x := g(y), g(y) = g(a), a = y$
Décomposition	$x := g(y), y = a, a = y$
Élimination de y	$x := g(a), y := a, a = a$
Suppression	$x := g(a), y := a$ solution

Unification : l'algorithme (exemple 5.3.11)

1. Résoudre $f(x, x, x) = f(g(y), g(a), y)$.

Décomposition	$x = g(y), x = g(a), x = y$
Élimination de x	$x := g(y), g(y) = g(a), g(y) = y$
Orientation	$x := g(y), g(y) = g(a), y = g(y)$
Échec d'élimination	il n'y a pas de solution

Remarque : L'algorithme est correct et se termine pour tout système. (Preuves dans le poly)

Plan

Introduction

Forme clausale

Unification

Résolution au 1er ordre

Complétude

Conclusion

Trois règles (exemples)

1. **Factorisation**

$$\frac{P(x,x) \vee P(y,a) \vee Q(y)}{P(a,a) \vee Q(a)} \quad \text{unification}$$

2. **Copie**

$$\frac{P(x,y)}{P(u,v)}$$

3. **Résolution binaire**

$$\frac{Q(x) \vee P(x,a) \qquad \neg P(b,y) \vee R(f(y))}{Q(b) \vee R(f(a))} \quad \text{unification}$$

Factorisation

Définition 5.4.2

La clause C' est un facteur de la clause C si :

- $C' = C$
- ou $C' = C\sigma$
  avec σ l'unificateur le plus général d'au moins 2 littéraux de C

Exemple 5.4.3

La clause $\underline{P(x)} \vee Q(g(x,y)) \vee \underline{P(f(a))}$ a deux facteurs :

- elle-même
- $P(f(a)) \vee Q(g(f(a),y))$ obtenu en appliquant $x := f(a)$

Propriété 5.4.4

Soit C' un facteur de la clause C : alors $\forall(C) \models \forall(C')$.
Preuve : En fait $\forall(A) \models \forall(A\sigma)$ pour toute formule A et toute substitution σ.

Copie

Définition 5.4.5

Soit σ une substitution qui :

- transforme toutes les variables en variables ;
- est une bijection.

La clause $C\sigma$ est une copie de la clause C.

σ est aussi appelée un renommage de C.

Exemple 5.4.7

Soit $\sigma = < x := u, y := v >$.
Le littéral $P(u, v)$ est une copie de $P(x, y)$.

Notons que $P(x, y)$ est aussi une copie de $P(u, v)$
par le renommage $\tau = < u := x, v := y >$ inverse de σ.

Copie

Propriété 5.4.8

Si σ est un renommage de C, alors C est aussi une copie de $C\sigma$.

Preuve.

On montre facilement que σ^{-1} est un renommage de $C\sigma$. □

Propriété 5.4.9

Si C et C' sont copies l'une de l'autre, alors $\forall(C) \equiv \forall(C')$.

Preuve.

C et C' sont instances l'une de l'autre.
Donc $\forall(C) \models \forall(C')$ et réciproquement. □

Résolvant binaire (RB)

Définition 5.4.10

Soient C et D deux clauses n'ayant pas de variable commune.
S'il y a deux littéraux :

- $L \in C$
- $M \in D$
- tels que L et M^c sont unifiables
- σ est la solution la plus générale de l'équation $L = M^c$

$E = ((C - \{L\}) \cup (D - \{M\}))\sigma$ est un résolvant binaire de C et D.

Résolvant binaire

Exemple 5.4.11

Soient $C = P(x, y) \vee P(y, k(z))$ et $D = \neg P(a, f(a, y_1))$.

$< x := a, y := f(a, y_1) >$ est la solution la plus générale de $P(x, y) = P(a, f(a, y_1))$.
Le (seul) résolvant binaire est $P(f(a, y_1), k(z))$.

Propriété 5.4.12

Soit E un résolvant binaire des clauses C et D : $\forall(C), \forall(D) \models \forall(E)$.

Résolution

Définition 5.4.13

Une preuve de C à partir de Γ est une suite de clauses toutes :

- élément de Γ,
- ou facteur d'une clause précédente,
- ou copie d'une clause précédente,
- ou résolvant binaire de 2 clauses précédentes,

se terminant par C.

On note $\Gamma \vdash_{1fcb} C$ s'il y a une preuve de C à partir de Γ.

Propriété 5.4.14 : cohérence

Si $\Gamma \vdash_{1fcb} C$ alors $\forall(\Gamma) \models \forall(C)$

Par récurrence, en se basant sur la cohérence des 3 règles.

Résolution : Exemple 5.4.15

Soient les deux clauses

1. $C_1 = P(x,y) \vee P(y,x)$
2. $C_2 = \neg P(u,z) \vee \neg P(z,u)$

Montrons par résolution que $\forall(C_1, C_2)$ n'a pas de modèle.

1.	$P(x,y) \vee P(y,x)$	Hyp C_1
2.	$P(y,y)$	Facteur de 1 $< x := y >$
3.	$\neg P(u,z) \vee \neg P(z,u)$	Hyp C_2
4.	$\neg P(z,z)$	Facteur de 3 $< u := z >$
5.	$\perp$	Résolvant Binaire 2, 4 $< y := z >$

Cet exemple montre que la résolution binaire seule est incomplète : sans la factorisation on ne peut pas déduire la clause vide.

Résolution : Exemple 5.4.16

1. $C_1 = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
2. $C_2 = P(z,f(z)) \vee P(z,a)$
3. $C_3 = P(f(z),z) \vee P(z,a)$

1.	$\neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$	Hyp C_1
2.	$P(z,f(z)) \vee P(z,a)$	Hyp C_2
3.	$P(v,f(v)) \vee P(v,a)$	Copie 2 $< z := v >$
4.	$\neg P(f(v),a) \vee \neg P(f(v),v) \vee P(v,a)$	RB 1(3), 3(1) $<z{:=}f(v)\,;\, x{:=}v>$
5.	$\neg P(f(a),a) \vee P(a,a)$	Fact 4 $< v := a >$
6.	$P(f(z),z) \vee P(z,a)$	Hyp C_3
7.	$P(a,a)$	RB 5(1), 6(1) $< z := a >$
8.	$\neg P(a,a)$	Fact 1 $< x := a; z := a >$
9.	$\perp$	RB 7, 8

Plan

Introduction

Forme clausale

Unification

Résolution au 1er ordre

Complétude

Conclusion

Résolution au 1° ordre

Dans ce système une seule règle, la résolution au 1° ordre, qui est une combinaison de factorisation, copie et résolution binaire.

Définition 5.4.17

La clause E est un résolvant au 1° ordre de C et D si :

- E est un résolvant binaire de C' et D', où
- C' est un facteur de C
- D' est une copie d'un facteur de D sans variable commune à C'

$$\frac{C \qquad D}{E}$$ est la règle de résolution au 1° ordre.

Exemple 5.4.18

Soient $C = \neg P(z,a) \vee \neg P(z,x) \vee \neg P(x,z)$
et $D = P(z,f(z)) \vee P(z,a)$.

- $C' = \neg P(a,a)$ est un facteur de C.
- D est facteur de lui-même (sans variable commune avec C').
- $P(a,f(a))$ est un résolvant binaire de C' et D.

Donc $P(a,f(a))$ est un résolvant au 1° ordre de C et D.

Trois notions de preuve par résolution

Soit à démontrer une clause C à partir d'un ensemble Γ :

Notations

1. $\Gamma \vdash_p C$: preuve par résolution propositionnelle (sans substitution).
2. $\Gamma \vdash_{1r} C$: preuve par résolution de 1° ordre.
3. $\Gamma \vdash_{1fcb} C$: preuve par factorisation, copie et résolution binaire.

Par définition, si $\Gamma \vdash_{1r} C$ alors $\Gamma \vdash_{1fcb} C$

Théorème du relèvement (1/3)

Théorème 5.4.19

Soient C' et D' des instances respectives de C et D.
Soit E' un résolvant propositionnel de C' et D'.

Alors E' est une instance d'un résolvant E au 1° ordre de C et D.

Exemple 5.4.20

$C = P(x) \vee P(y) \vee R(y)$ et $D = \neg Q(x) \vee P(x) \vee \neg R(x) \vee P(y)$.

- $C' = P(a) \vee R(a)$ et $D' = \neg Q(a) \vee P(a) \vee \neg R(a)$ en sont des instances.
- $E' = P(a) \vee \neg Q(a)$ est un résolvant propositionnel de C' et D'.
- $E = P(x) \vee \neg Q(x)$ est un résolvant au 1° ordre de C et D tel que $E' = E < x := a >$.

Théorème du relèvement (2/3)

Théorème 5.4.21

Soit Δ un ensemble d'instances des clauses de Γ.
Soit $C_1, \ldots, C_n$ une preuve par résolution propositionnelle à partir de Δ.

Il existe une preuve $D_1, \ldots, D_n$ par résolution de 1° ordre à partir de Γ telle que chaque C_i est une instance de D_i.

Preuve.

Par récurrence. Soit $C_1 \ldots C_{n+1}$ une preuve par résolution prop. à partir de Δ.
Par hypothèse de récurrence, on a une preuve $D_1, \ldots, D_n$ au 1° ordre.

1. Si $C_{n+1} \in \Delta$, alors C_{n+1} est instance d'une clause de Γ : c'est D_{n+1}.
2. Si C_{n+1} est un résolvant propositionnel de C_j et C_k, on prend le résolvant au 1° ordre de D_j et D_k obtenu par le théorème précédent.

$\square$

Théorème du relèvement (3/3)

Corollaire 5.4.22

Soient Γ un ensemble de clauses
et Δ un ensemble d'instances des clauses de Γ.

Supposons que $\Delta \vdash_p C$.

Alors il existe D telle que :

- $\Gamma \vdash_{1r} D$
- C est une instance de D.

On dit que la preuve de C est relevée en une preuve au premier ordre.

Exemple 5.4.23

$\Gamma = \{P(f(x)) \vee P(u), \neg P(x) \vee Q(z), \neg Q(x) \vee \neg Q(y)\}$.

$\forall(\Gamma)$ est insatisfaisable et nous le montrons de trois manières.

1. Instanciation sur le domaine de Herbrand $a, f(a), f(f(a)), \ldots$:

$$\begin{array}{rcl} P(f(x)) \vee P(u) & \text{est instanciée en} & P(f(a)) \\ \neg P(x) \vee Q(z) & \text{est instanciée en} & \neg P(f(a)) \vee Q(a) \\ \neg Q(x) \vee \neg Q(y) & \text{est instanciée en} & \neg Q(a) \end{array}$$

Preuve par $\vdash_p$ que cet ensemble d'instances est insatisfaisable :

$$\dfrac{\dfrac{P(f(a)) \qquad \neg P(f(a)) \vee Q(a)}{Q(a)} \qquad \neg Q(a)}{\bot}$$

Exemple 5.4.23

$$\Gamma = \{P(f(x)) \vee P(u), \neg P(x) \vee Q(z), \neg Q(x) \vee \neg Q(y)\}$$

1. Preuve par instanciation et $\vdash_p$

$$\dfrac{\dfrac{P(f(a)) \qquad \neg P(f(a)) \vee Q(a)}{Q(a)} \qquad \neg Q(a)}{\bot}$$

2. On relève cette preuve par $\vdash_p$ en une preuve par $\vdash_{1r}$:

$$\dfrac{\dfrac{P(f(x)) \vee P(u) \qquad \neg P(x) \vee Q(z)}{Q(z)} \qquad \neg Q(x) \vee \neg Q(y)}{\bot}$$

Exemple 5.4.23

2. Preuve par $\vdash_{1r}$

$$\dfrac{\dfrac{P(f(x)) \vee P(u) \qquad \neg P(x) \vee Q(z)}{Q(z)} \qquad \neg Q(x) \vee \neg Q(y)}{\bot}$$

3. Chaque règle de $\vdash_{1r}$ est décomposée en factorisation, copie et résolution binaire :

$$\dfrac{\dfrac{\dfrac{P(f(x)) \vee P(u)}{P(f(x))}\,fact \qquad \dfrac{\neg P(x) \vee Q(z)}{\neg P(y) \vee Q(z)}\,copie}{Q(z)}\,rb \qquad \dfrac{\neg Q(x) \vee \neg Q(y)}{\neg Q(x)}\,fact}{\bot}\,rb$$

Complétude réfutationnelle de la résolution au 1° ordre

Théorème 5.4.24

Les trois propositions
1. $\Gamma \vdash_{1r} \bot$
2. $\Gamma \vdash_{1fcb} \bot$
3. $\forall(\Gamma) \models \bot$

sont équivalentes.

Démonstration.

- $(1 \Rightarrow 2)$ car une résolution au 1° ordre se décompose en factorisation, copie et résolution binaire.
- $(2 \Rightarrow 3)$ par cohérence de la factorisation, copie et résolution binaire.
- $(3 \Rightarrow 1)$. Supposons que $\forall(\Gamma)$ est insatisfaisable.

  D'après Herbrand, on a un ensemble fini insatisfaisable d'instances Δ.

  Par complétude de la résolution propositionnelle, on a $\Delta \vdash_p \bot$.

  Par relèvement, $\Gamma \vdash_{1r} D$ où $\bot$ est une instance de D, d'où $D = \bot$.

Preuves automatiques

Pour produire automatiquement des preuves en résolution binaire, il est possible d'utiliser le logiciel (principe similaire à la *stratégie complète*) :

http://teachinglogic.univ-grenoble-alpes.fr/ResBinSc/

Si l'ensemble de clauses est insatisfaisable, le logiciel est théoriquement capable de déduire la clause vide (en un temps illimité).

Que peut-on en conclure ?

- si le logiciel affirme qu'il a déduit la clause vide :
  - les clauses sont effectivement insatisfaisables
  - il en fournit une preuve
- si le logiciel affirme qu'il ne peut pas prouver la clause vide ou s'il atteint sa limite de temps :
  - on ne peut rien conclure

Plan

Introduction

Forme clausale

Unification

Résolution au 1er ordre

Complétude

Conclusion

Aujourd'hui

- L'unification permet de déterminer des instanciations judicieuses de clauses avec variables
- La résolution au premier ordre intègre dans un même système déductif la **recherche d'instances insatisfaisables** et la **preuve d'insatisfaisabilité** d'un ensemble de clauses
- Le relèvement permet d'obtenir une preuve au premier ordre à partir d'une preuve propositonnelle, et a pour corollaire la **complétude** de la résolution au premier ordre (qui est également **correcte**).

Plan du Semestre

- Logique propositionnelle
- Résolution propositionnelle
- Déduction naturelle propositionnelle

PARTIEL

- Logique du premier ordre
- Base de la démonstration automatique (“résolution au premier ordre”) *
- Déduction naturelle au premier ordre

EXAMEN

La prochaine fois

Déduction naturelle au premier ordre

- Règles
- Exemples

Déduction Naturelle: quantificateurs, copie et égalité

Benjamin Wack

Université Grenoble Alpes

Avril 2023

Motivation

Cas propositionnel

Il y a des algorithmes pour décider si une formule est valide ou non.

Cas du premier ordre

Il n'y a aucun algorithme pour décider si une formule est valide ou non.

Motivation

Cas propositionnel

Il y a des algorithmes pour décider si une formule est valide ou non.

Cas du premier ordre

Il n'y a aucun algorithme pour décider si une formule est valide ou non.

En admettant l'équivalence entre prouvable et valide, il n'y a pas d'algorithme qui, étant donné une formule du 1° ordre, puisse :

- en construire une preuve
- ou nous avertir que cette formule n'a pas de preuve.

Alonzo Church (1903-1995), logicien américain

- Inventeur du lambda-calcul (1936)
  $(\lambda x.xy)\ (\lambda z.z)\ \rightarrow_\beta\ (\lambda z.z)y$
  - proposition de modèle de calcul universel
  - à la base des langages fonctionnels (ML, Lisp...)
  - permet de représenter des programmes aussi bien que des démonstrations
  - une des premières notions de typage

Alonzo Church (1903-1995), logicien américain

- Inventeur du lambda-calcul (1936)
  $(\lambda x.xy)\ (\lambda z.z)\ \rightarrow_\beta\ (\lambda z.z)y$
  - proposition de modèle de calcul universel
  - à la base des langages fonctionnels (ML, Lisp...)
  - permet de représenter des programmes aussi bien que des démonstrations
  - une des premières notions de typage
- Preuve que la logique du premier ordre est indécidable algorithmiquement (remise en cause du programme de Hilbert)
- Résultats redémontrés indépendemment par Turing (1937)

Alonzo Church (1903-1995), logicien américain

- Inventeur du lambda-calcul (1936)
  $(\lambda x.xy)\ (\lambda z.z)\ \rightarrow_\beta\ (\lambda z.z)y$
  - proposition de modèle de calcul universel
  - à la base des langages fonctionnels (ML, Lisp...)
  - permet de représenter des programmes aussi bien que des démonstrations
  - une des premières notions de typage
- Preuve que la logique du premier ordre est indécidable algorithmiquement (remise en cause du programme de Hilbert)
- Résultats redémontrés indépendemment par Turing (1937)
- Thèse de Church-Turing : le λ-calcul ou la machine de Turing expriment exactement ce qu'est un calcul mécanique

Plan

Introduction

Règles et exemples

Règle de la copie

Les règles de l'égalité

Tactiques de preuves

Conclusion

Plan

Introduction

Règles et exemples

Règle de la copie

Les règles de l'égalité

Tactiques de preuves

Conclusion

Rappel : Règles « propositionnelles »

Table 3.1

Introduction	Élimination
$\dfrac{\begin{array}{c}[A]\\ \cdots \\ B\end{array}}{A \Rightarrow B} \Rightarrow I$	$\dfrac{A \quad A \Rightarrow B}{B} \Rightarrow E$
$\dfrac{A \quad B}{A \wedge B} \wedge I$	$\dfrac{A \wedge B}{A} \wedge E1$ $\dfrac{A \wedge B}{B} \wedge E2$
$\dfrac{A}{A \vee B} \vee I1$ $\dfrac{A}{B \vee A} \vee I2$	$\dfrac{A \vee B \quad A \Rightarrow C \quad B \Rightarrow C}{C} \vee E$
Règle du faux	
$\dfrac{\bot}{A}\ Efq$	
Réduction à l'absurde	
$\dfrac{\neg\neg A}{A}\ RAA$	

Extension de la déduction naturelle propositionnelle

- Les définitions de **brouillon de preuve, environnement, contexte, formule utilisable** restent inchangées !

Extension de la déduction naturelle propositionnelle

- Les définitions de **brouillon de preuve, environnement, contexte, formule utilisable** restent inchangées !
- Une seule règle pour enlever des hypothèses : $\Rightarrow$ I.

Extension de la déduction naturelle propositionnelle

- Les définitions de **brouillon de preuve, environnement, contexte, formule utilisable** restent inchangées !
- Une seule règle pour enlever des hypothèses : $\Rightarrow$ I.

Règles en plus par rapport à la DN « propositionnelle ».

- les quantificateurs
- la copie
- l'égalité

Cohérence et complétude

Cohérence et complétude

- **Cohérence :** $\Gamma \vdash A$ implique $\Gamma \models A$.

  Admise.
  L'essentiel est de montrer que les nouvelles règles sont correctes.

Cohérence et complétude

- **Cohérence :** $\Gamma \vdash A$ implique $\Gamma \models A$.

  Admise.
  L'essentiel est de montrer que les nouvelles règles sont correctes.

- **Complétude :** $\Gamma \models A$ implique $\Gamma \vdash A$.
  Admise sans preuve.

Plan

Introduction

Règles et exemples

Règle de la copie

Les règles de l'égalité

Tactiques de preuves

Conclusion

Introduction

Une règle d'élimination et une règle d'introduction pour chaque quantificateur.

- Comment utiliser ces règles sur des exemples.
- Ainsi que les erreurs occasionnées par le non respect des conditions d'emploi des règles.

Rappel

Définition 4.3.34

Soit x une variable, t un terme et A une formule.

1. $A < x := t >$ est la formule obtenue en remplaçant dans la formule A toute occurrence libre de x par le terme t.
2. Le terme t est libre pour x dans A si les variables de t ne sont pas liées dans les occurrences libres de x.

Rappel

Définition 4.3.34

Soit x une variable, t un terme et A une formule.

1. $A < x := t >$ est la formule obtenue en remplaçant dans la formule A toute occurrence libre de x par le terme t.
2. Le terme t est libre pour x dans A si les variables de t ne sont pas liées dans les occurrences libres de x.

Exemple

$$A = \forall y P(x, y)$$

- z est-il libre pour x dans A ?

Rappel

Définition 4.3.34

Soit x une variable, t un terme et A une formule.

1. $A < x := t >$ est la formule obtenue en remplaçant dans la formule A toute occurrence libre de x par le terme t.
2. Le terme t est libre pour x dans A si les variables de t ne sont pas liées dans les occurrences libres de x.

Exemple

$$A = \forall y P(x, y)$$

- z est-il libre pour x dans A? oui
- $g(y)$ est-il libre pour x dans A?

Rappel

Définition 4.3.34

Soit x une variable, t un terme et A une formule.

1. $A < x := t >$ est la formule obtenue en remplaçant dans la formule A toute occurrence libre de x par le terme t.
2. Le terme t est libre pour x dans A si les variables de t ne sont pas liées dans les occurrences libres de x.

Exemple

$$A = \forall y P(x, y)$$

- z est-il libre pour x dans A? oui
- $g(y)$ est-il libre pour x dans A? non
- x est-il libre pour y dans A?

Rappel

Définition 4.3.34

Soit x une variable, t un terme et A une formule.

1. $A < x := t >$ est la formule obtenue en remplaçant dans la formule A toute occurrence libre de x par le terme t.
2. Le terme t est libre pour x dans A si les variables de t ne sont pas liées dans les occurrences libres de x.

Exemple

$$A = \forall y P(x, y)$$

- z est-il libre pour x dans A? oui
- $g(y)$ est-il libre pour x dans A? non
- x est-il libre pour y dans A? oui

Règles des quantificateurs : $\forall$E

A et B sont des formules, x est une variable, t est un terme

Elimination $\forall$

$$\frac{\forall x A}{A < x := t >} \forall E$$

t doit être libre pour x dans A.

Exemple 6.1.1

Usage incorrect de la règle ∀E : où est l'erreur ?

1	1	supposons $\forall x \exists y P(x,y)$	
1	2	$\exists y P(y,y)$	$\forall E$ 1, y
	3	donc $\forall x \exists y P(x,y) \Rightarrow \exists y P(y,y)$	

Exemple 6.1.1

Usage incorrect de la règle ∀E : où est l'erreur ?

1	1	supposons $\forall x \exists y P(x,y)$	
1	2	$\exists y P(y,y)$	$\forall E$ 1, y ERREUR
	3	donc $\forall x \exists y P(x,y) \Rightarrow \exists y P(y,y)$	

À la ligne 2, on n'a pas respecté les conditions d'applications de ∀E car le terme y n'est pas libre pour x dans la formule $\exists y P(x,y)$.

Exemple 6.1.1

Usage incorrect de la règle ∀E : où est l'erreur ?

1	1	supposons $\forall x \exists y P(x,y)$	
1	2	$\exists y P(y,y)$	$\forall E$ 1, y ERREUR
	3	donc $\forall x \exists y P(x,y) \Rightarrow \exists y P(y,y)$	

À la ligne 2, on n'a pas respecté les conditions d'applications de ∀E car le terme y n'est pas libre pour x dans la formule $\exists y P(x,y)$.

Soit I l'interprétation de domaine $\{0,1\}$ avec $P_I =$

$$\boxed{\{(0,1),(1,0)\}}$$

Cette interprétation rend fausse la « conclusion ».

Règles des quantificateurs : $\forall$ I

A et B sont des formules, x est une variable.

Introduction $\forall$

$$\frac{A}{\forall x A}\forall I$$

x ne doit être libre

- ni dans l'**environnement** de la preuve,
- ni dans le **contexte** de la prémisse de la règle.

Exemple 6.1.2 $\forall y P(y) \wedge \forall y Q(y) \Rightarrow \forall x (P(x) \wedge Q(x))$

Exemple 6.1.2 $\forall y P(y) \wedge \forall y Q(y) \Rightarrow \forall x (P(x) \wedge Q(x))$

1	1	supposons $\forall y P(y) \wedge \forall y Q(y)$

Exemple 6.1.2 $\forall y P(y) \wedge \forall y Q(y) \Rightarrow \forall x (P(x) \wedge Q(x))$

1	1	supposons $\forall y P(y) \wedge \forall y Q(y)$	
1	2	$\forall y P(y)$	$\wedge E1$ 1

Exemple 6.1.2 $\forall y P(y) \wedge \forall y Q(y) \Rightarrow \forall x (P(x) \wedge Q(x))$

1	1	supposons $\forall y P(y) \wedge \forall y Q(y)$	
1	2	$\forall y P(y)$	$\wedge E1$ 1
1	3	$\forall y Q(y)$	$\wedge E2$ 1

Exemple 6.1.2 $\forall y P(y) \wedge \forall y Q(y) \Rightarrow \forall x (P(x) \wedge Q(x))$

1	1	supposons $\forall y P(y) \wedge \forall y Q(y)$	
1	2	$\forall y P(y)$	$\wedge E1$ 1
1	3	$\forall y Q(y)$	$\wedge E2$ 1
1	4	$P(x)$	$\forall E$ 2, x
1	5	$Q(x)$	$\forall E$ 3, x

Remarque : Dans l'usage de la règle $\forall$E aux lignes 4 et 5, on précise que y est remplacé par x.

Exemple 6.1.2 $\forall y P(y) \wedge \forall y Q(y) \Rightarrow \forall x (P(x) \wedge Q(x))$

1	1	supposons $\forall y P(y) \wedge \forall y Q(y)$	
1	2	$\forall y P(y)$	$\wedge E1$ 1
1	3	$\forall y Q(y)$	$\wedge E2$ 1
1	4	$P(x)$	$\forall E$ 2, x
1	5	$Q(x)$	$\forall E$ 3, x
1	6	$P(x) \wedge Q(x)$	$\wedge I$ 4, 5

Remarque : Dans l'usage de la règle $\forall$E aux lignes 4 et 5, on précise que y est remplacé par x.

Exemple 6.1.2 $\forall y P(y) \wedge \forall y Q(y) \Rightarrow \forall x (P(x) \wedge Q(x))$

1	1	supposons $\forall y P(y) \wedge \forall y Q(y)$	
1	2	$\forall y P(y)$	$\wedge E1$ 1
1	3	$\forall y Q(y)$	$\wedge E2$ 1
1	4	$P(x)$	$\forall E$ 2, x
1	5	$Q(x)$	$\forall E$ 3, x
1	6	$P(x) \wedge Q(x)$	$\wedge I$ 4, 5
1	7	$\forall x (P(x) \wedge Q(x))$	$\forall I$ 6

Remarque : Dans l'usage de la règle ∀E aux lignes 4 et 5, on précise que y est remplacé par x.

Exemple 6.1.2 $\forall y P(y) \wedge \forall y Q(y) \Rightarrow \forall x (P(x) \wedge Q(x))$

1	1	supposons $\forall y P(y) \wedge \forall y Q(y)$	
1	2	$\forall y P(y)$	$\wedge E1$ 1
1	3	$\forall y Q(y)$	$\wedge E2$ 1
1	4	$P(x)$	$\forall E$ 2, x
1	5	$Q(x)$	$\forall E$ 3, x
1	6	$P(x) \wedge Q(x)$	$\wedge I$ 4, 5
1	7	$\forall x (P(x) \wedge Q(x))$	$\forall I$ 6
	8	donc $\forall y P(y) \wedge \forall y Q(y) \Rightarrow \forall x (P(x) \wedge Q(x))$	$\Rightarrow$I 1, 7

Remarque : Dans l'usage de la règle $\forall$E aux lignes 4 et 5, on précise que y est remplacé par x.

Exemple 6.1.3

Usage incorrect de la règle $\forall$I

1	1	supposons $P(x)$	
1	2	$\forall x P(x)$	$\forall I$ 1
	3	donc $P(x) \Rightarrow \forall x P(x)$	$\Rightarrow$I 1, 2

Exemple 6.1.3

Usage incorrect de la règle $\forall$I

1	1	supposons $P(x)$	
1	2	$\forall x P(x)$	$\forall I$ 1 ERREUR
	3	donc $P(x) \Rightarrow \forall x P(x)$	$\Rightarrow$I 1, 2

À la ligne 2, x est libre dans le contexte $P(x)$, ce qui interdit de généraliser sur x.

Exemple 6.1.3

Usage incorrect de la règle $\forall$I

1	1	supposons $P(x)$	
1	2	$\forall x P(x)$	$\forall I$ 1 ERREUR
	3	donc $P(x) \Rightarrow \forall x P(x)$	$\Rightarrow$I 1, 2

À la ligne 2, x est libre dans le contexte $P(x)$, ce qui interdit de généraliser sur x.

Soit I l'interprétation de domaine $\{0,1\}$ avec $P_I = \{0\}$.
Soit e un état où $x = 0$.
L'assignation (I, e) rend fausse la « conclusion ».

Règles des quantificateurs : ∃E

A et B sont des formules, x est une variable.

Elimination ∃

$$\frac{\exists x A \qquad (A \Rightarrow B)}{B}\exists E$$

x ne doit être libre

- ni dans l'environnement,
- ni dans B,
- ni dans le contexte de $A \Rightarrow B$.

Exemple 6.1.4

Usage incorrect de la règle ∃E

1	1	Supposons $\exists x P(x) \wedge (P(x) \Rightarrow \forall y Q(y))$	
1	2	$\exists x P(x)$	$\wedge E1$ 1
1	3	$P(x) \Rightarrow \forall y Q(y)$	$\wedge E2$ 1
1	4	$\forall y Q(y)$	$\exists E$ 2, 3
	5	Donc $\exists x P(x) \wedge (P(x) \Rightarrow \forall y Q(y)) \Rightarrow \forall y Q(y)$	$\Rightarrow I$ 1,4

Exemple 6.1.4

Usage incorrect de la règle ∃E

1	1	Supposons $\exists x P(x) \wedge (P(x) \Rightarrow \forall y Q(y))$	
1	2	$\exists x P(x)$	$\wedge E1$ 1
1	3	$P(x) \Rightarrow \forall y Q(y)$	$\wedge E2$ 1
1	4	$\forall y Q(y)$	$\exists E$ 2, 3 ERREUR
	5	Donc $\exists x P(x) \wedge (P(x) \Rightarrow \forall y Q(y)) \Rightarrow \forall y Q(y)$	$\Rightarrow I$ 1,4

Le contexte de la prémisse $P(x) \Rightarrow \forall y Q(y)$ ne doit pas dépendre de x.

Exemple 6.1.4

Usage incorrect de la règle ∃E

1	1	Supposons $\exists x P(x) \wedge (P(x) \Rightarrow \forall y Q(y))$	
1	2	$\exists x P(x)$	$\wedge E1$ 1
1	3	$P(x) \Rightarrow \forall y Q(y)$	$\wedge E2$ 1
1	4	$\forall y Q(y)$	$\exists E$ 2, 3 ERREUR
	5	Donc $\exists x P(x) \wedge (P(x) \Rightarrow \forall y Q(y)) \Rightarrow \forall y Q(y)$	$\Rightarrow I$ 1,4

Le contexte de la prémisse $P(x) \Rightarrow \forall y Q(y)$ ne doit pas dépendre de x.

Soit I l'interprétation de domaine $\{0, 1\}$ avec $P_I = Q_I = \{0\}$.
Soit l'état e où $x = 1$.
L'assignation (I, e) rend fausse cette « conclusion ».

Exemple 6.1.5

Usage incorrect de la règle ∃E

1	1	supposons $\exists x P(x)$	
1, 2	2	supposons $P(x)$	
1	3	donc $P(x) \Rightarrow P(x)$	$\Rightarrow I$ 2, 2
1	4	$P(x)$	$\exists E$ 1, 3
1	5	$\forall x P(x)$	$\forall I$ 4
	6	donc $\exists x P(x) \Rightarrow \forall x P(x)$	

Exemple 6.1.5

Usage incorrect de la règle ∃E

1	1	supposons $\exists x P(x)$	
1, 2	2	supposons $P(x)$	
1	3	donc $P(x) \Rightarrow P(x)$	$\Rightarrow I$ 2, 2
1	4	$P(x)$	$\exists E$ 1, 3 ERREUR
1	5	$\forall x P(x)$	$\forall I$ 4
	6	donc $\exists x P(x) \Rightarrow \forall x P(x)$	

La conclusion de la règle $\exists E$ ne doit pas dépendre de x.

Exemple 6.1.5

Usage incorrect de la règle ∃E

1	1	supposons $\exists x P(x)$	
1, 2	2	supposons $P(x)$	
1	3	donc $P(x) \Rightarrow P(x)$	$\Rightarrow I$ 2, 2
1	4	$P(x)$	$\exists E$ 1, 3 ERREUR
1	5	$\forall x P(x)$	$\forall I$ 4
	6	donc $\exists x P(x) \Rightarrow \forall x P(x)$	

La conclusion de la règle $\exists E$ ne doit pas dépendre de x.

Soit I l'interprétation de domaine $\{0, 1\}$ avec $P_I = \{0\}$.
I rend fausse la « conclusion ».

Règles des quantificateurs : $\exists$I

A et B sont des formules, x est une variable, t est un terme

Introduction $\exists$

$$\frac{A < x := t >}{\exists x A} \exists I$$

t doit être libre pour x dans A.

Exemple 6.1.6 $\neg\forall x A \Rightarrow \exists x \neg A$ (loi de De Morgan)

Exemple 6.1.6 $\neg\forall xA \Rightarrow \exists x\neg A$ (loi de De Morgan)

1	1	Supposons $\neg\forall xA$

Exemple 6.1.6 $\neg\forall xA \Rightarrow \exists x\neg A$ (loi de De Morgan)

1	1	Supposons $\neg\forall xA$
1, 2	2	Supposons $\neg\exists x\neg A$

Exemple 6.1.6 $\neg\forall xA \Rightarrow \exists x\neg A$ (loi de De Morgan)

1	1	Supposons $\neg\forall xA$
1, 2	2	Supposons $\neg\exists x\neg A$
1, 2, 3	3	Supposons $\neg A$

Exemple 6.1.6 $\neg\forall xA \Rightarrow \exists x\neg A$ (loi de De Morgan)

1	1	Supposons $\neg\forall xA$	
1, 2	2	Supposons $\neg\exists x\neg A$	
1, 2, 3	3	Supposons $\neg A$	
1, 2, 3	4	$\exists x\neg A$	$\exists I$ 3, x

- À la ligne 4 : on utilise $\neg A = \neg A < x := x >$
  et une variable x est toujours libre pour elle-même dans A.

Exemple 6.1.6 $\neg\forall xA \Rightarrow \exists x\neg A$ (loi de De Morgan)

1	1	Supposons $\neg\forall xA$	
1, 2	2	Supposons $\neg\exists x\neg A$	
1, 2, 3	3	Supposons $\neg A$	
1, 2, 3	4	$\exists x\neg A$	$\exists I$ 3, x
1, 2, 3	5	$\perp$	$\Rightarrow E$ 2, 4

- À la ligne 4 : on utilise $\neg A = \neg A < x := x >$
  et une variable x est toujours libre pour elle-même dans A.

Exemple 6.1.6 $\neg\forall xA \Rightarrow \exists x\neg A$ (loi de De Morgan)

1	1	Supposons $\neg\forall xA$	
1, 2	2	Supposons $\neg\exists x\neg A$	
1, 2, 3	3	Supposons $\neg A$	
1, 2, 3	4	$\exists x\neg A$	$\exists I$ 3, x
1, 2, 3	5	$\perp$	$\Rightarrow E$ 2, 4
1, 2	6	Donc $\neg\neg A$	$\Rightarrow I$ 3, 5

- À la ligne 4 : on utilise $\neg A = \neg A < x := x >$
  et une variable x est toujours libre pour elle-même dans A.

Exemple 6.1.6 $\neg\forall xA \Rightarrow \exists x\neg A$ (loi de De Morgan)

1	1	Supposons $\neg\forall xA$	
1, 2	2	Supposons $\neg\exists x\neg A$	
1, 2, 3	3	Supposons $\neg A$	
1, 2, 3	4	$\exists x\neg A$	$\exists I$ 3, x
1, 2, 3	5	$\perp$	$\Rightarrow E$ 2, 4
1, 2	6	Donc $\neg\neg A$	$\Rightarrow I$ 3, 5
1, 2	7	A	Raa 6

- À la ligne 4 : on utilise $\neg A = \neg A < x := x >$
  et une variable x est toujours libre pour elle-même dans A.

Exemple 6.1.6 $\neg\forall xA \Rightarrow \exists x\neg A$ (loi de De Morgan)

1	1	Supposons $\neg\forall xA$	
1, 2	2	Supposons $\neg\exists x\neg A$	
1, 2, 3	3	Supposons $\neg A$	
1, 2, 3	4	$\exists x\neg A$	$\exists I$ 3, x
1, 2, 3	5	$\perp$	$\Rightarrow E$ 2, 4
1, 2	6	Donc $\neg\neg A$	$\Rightarrow I$ 3, 5
1, 2	7	A	Raa 6
1, 2	8	$\forall xA$	$\forall I$ 7

- À la ligne 4 : on utilise $\neg A = \neg A < x := x >$
  et une variable x est toujours libre pour elle-même dans A.

Exemple 6.1.6 $\neg\forall xA \Rightarrow \exists x\neg A$ (loi de De Morgan)

1	1	Supposons $\neg\forall xA$	
1, 2	2	Supposons $\neg\exists x\neg A$	
1, 2, 3	3	Supposons $\neg A$	
1, 2, 3	4	$\exists x\neg A$	$\exists I$ 3, x
1, 2, 3	5	$\perp$	$\Rightarrow E$ 2, 4
1, 2	6	Donc $\neg\neg A$	$\Rightarrow I$ 3, 5
1, 2	7	A	Raa 6
1, 2	8	$\forall xA$	$\forall I$ 7
1, 2	9	$\perp$	$\Rightarrow E$ 1, 8

- À la ligne 4 : on utilise $\neg A = \neg A < x := x >$
  et une variable x est toujours libre pour elle-même dans A.

Exemple 6.1.6 $\neg\forall xA \Rightarrow \exists x\neg A$ (loi de De Morgan)

1	1	Supposons $\neg\forall xA$	
1, 2	2	Supposons $\neg\exists x\neg A$	
1, 2, 3	3	Supposons $\neg A$	
1, 2, 3	4	$\exists x\neg A$	$\exists I$ 3, x
1, 2, 3	5	$\perp$	$\Rightarrow E$ 2, 4
1, 2	6	Donc $\neg\neg A$	$\Rightarrow I$ 3, 5
1, 2	7	A	Raa 6
1, 2	8	$\forall xA$	$\forall I$ 7
1, 2	9	$\perp$	$\Rightarrow E$ 1, 8
1	10	Donc $\neg\neg\exists x\neg A$	$\Rightarrow$ I 2, 9

- À la ligne 4 : on utilise $\neg A = \neg A < x := x >$
  et une variable x est toujours libre pour elle-même dans A.

Exemple 6.1.6 $\neg\forall xA \Rightarrow \exists x\neg A$ (loi de De Morgan)

1	1	Supposons $\neg\forall xA$	
1, 2	2	Supposons $\neg\exists x\neg A$	
1, 2, 3	3	Supposons $\neg A$	
1, 2, 3	4	$\exists x\neg A$	$\exists I$ 3, x
1, 2, 3	5	$\perp$	$\Rightarrow E$ 2, 4
1, 2	6	Donc $\neg\neg A$	$\Rightarrow I$ 3, 5
1, 2	7	A	Raa 6
1, 2	8	$\forall xA$	$\forall I$ 7
1, 2	9	$\perp$	$\Rightarrow E$ 1, 8
1	10	Donc $\neg\neg\exists x\neg A$	$\Rightarrow$ I 2, 9
1	11	$\exists x\neg A$	Raa 10

- À la ligne 4 : on utilise $\neg A = \neg A < x := x >$
  et une variable x est toujours libre pour elle-même dans A.

Exemple 6.1.6 $\neg\forall xA \Rightarrow \exists x\neg A$ (loi de De Morgan)

1	1	Supposons $\neg\forall xA$	
1, 2	2	Supposons $\neg\exists x\neg A$	
1, 2, 3	3	Supposons $\neg A$	
1, 2, 3	4	$\exists x\neg A$	$\exists I$ 3, x
1, 2, 3	5	$\perp$	$\Rightarrow E$ 2, 4
1, 2	6	Donc $\neg\neg A$	$\Rightarrow I$ 3, 5
1, 2	7	A	Raa 6
1, 2	8	$\forall xA$	$\forall I$ 7
1, 2	9	$\perp$	$\Rightarrow E$ 1, 8
1	10	Donc $\neg\neg\exists x\neg A$	$\Rightarrow$ I 2, 9
1	11	$\exists x\neg A$	Raa 10
	12	Donc $\neg\forall xA \Rightarrow \exists x\neg A$	$\Rightarrow$ I 1, 11

- À la ligne 4 : on utilise $\neg A = \neg A < x := x >$
  et une variable x est toujours libre pour elle-même dans A.

Récapitulatif des règles des quantificateurs : Tableau 6.1

$\dfrac{A}{\forall x A}\ \forall I$	x ne doit être libre ▶ ni dans l'environnement de la preuve, ▶ ni dans le contexte de la prémisse
$\dfrac{\forall x A}{A<x:=t>}\ \forall E$	t doit être libre pour x dans A
$\dfrac{A<x:=t>}{\exists x A}\ \exists I$	t doit être libre pour x dans A
$\dfrac{\exists x A \qquad (A \Rightarrow B)}{B}\ \exists E$	x ne doit être libre ▶ ni dans l'environnement ▶ ni dans B, ▶ ni dans le contexte de $A \Rightarrow B$

Plan

Introduction

Règles et exemples

Règle de la copie

Les règles de l'égalité

Tactiques de preuves

Conclusion

Définition

La règle de copie consiste à déduire d'une formule, une autre formule égale au changement près des variables liées.

$$\frac{A'}{A}\ \text{copie}$$

Rappels : changement de variables liées (1/3)

Deux formules sont α-équivalentes si on peut obtenir l'une à partir de l'autre en remplaçant des sous-formules $Qx\ A$ par $Qy\ A < x := y >$ où Q est un quantificateur et y ne figure pas dans $Qx\ A$.

Rappels : changement de variables liées (1/3)

Deux formules sont α-équivalentes si on peut obtenir l'une à partir de l'autre en remplaçant des sous-formules $Qx\ A$ par $Qy\ A < x := y >$ où Q est un quantificateur et y ne figure pas dans $Qx\ A$.

Exemple 4.4.4

- $\forall x\ p(x,z) =_\alpha \forall y\ p(y,z)$.
- $\forall x\ p(x,z) \neq_\alpha \forall z\ p(z,z)$.

Changement de variables liées (2/3)

Définition 4.4.5

Deux formules sont égales à un changement près de variables liées si on obtient l'une à partir de l'autre par des remplacements de la forme

$$Qx\ A \equiv Qy\ A < x := y > \quad \text{où } y \text{ ne figure pas dans } Qx\ A$$

On dit que les deux formules sont :

- α-équivalentes
- copie l'une de l'autre
- $A =_\alpha B$

Changement de variables liées (3/3)

Théorème 4.4.6

Si deux formules sont égales à un changement près de variables liées alors elles sont équivalentes.

Exemple 4.4.7

Montrons que $\forall x \exists y P(x, y)$ et $\forall y \exists x P(y, x)$ sont équivalentes.

Changement de variables liées (3/3)

Théorème 4.4.6

Si deux formules sont égales à un changement près de variables liées alors elles sont équivalentes.

Exemple 4.4.7

Montrons que $\forall x \exists y P(x, y)$ et $\forall y \exists x P(y, x)$ sont équivalentes.

$$\forall x \exists y P(x, y)$$

Changement de variables liées (3/3)

Théorème 4.4.6

Si deux formules sont égales à un changement près de variables liées alors elles sont équivalentes.

Exemple 4.4.7

Montrons que $\forall x \exists y P(x, y)$ et $\forall y \exists x P(y, x)$ sont équivalentes.

$$\begin{aligned} & \forall x \exists y P(x, y) \\ =_\alpha \quad & \forall u \exists y P(u, y) \end{aligned}$$

Changement de variables liées (3/3)

Théorème 4.4.6

Si deux formules sont égales à un changement près de variables liées alors elles sont équivalentes.

Exemple 4.4.7

Montrons que $\forall x \exists y P(x, y)$ et $\forall y \exists x P(y, x)$ sont équivalentes.

$$\begin{aligned} & \forall x \exists y P(x, y) \\ =_\alpha \quad & \forall u \exists y P(u, y) \\ =_\alpha \quad & \forall u \exists x P(u, x) \end{aligned}$$

Changement de variables liées (3/3)

Théorème 4.4.6

Si deux formules sont égales à un changement près de variables liées alors elles sont équivalentes.

Exemple 4.4.7

Montrons que $\forall x \exists y P(x, y)$ et $\forall y \exists x P(y, x)$ sont équivalentes.

$$\begin{aligned} & \forall x \exists y P(x, y) \\ =_\alpha \quad & \forall u \exists y P(u, y) \\ =_\alpha \quad & \forall u \exists x P(u, x) \\ =_\alpha \quad & \forall y \exists x P(y, x) \end{aligned}$$

Savoir si deux formules sont α-équivalentes

Technique

- Tracer un trait entre chaque quantificateur et les variables qu'il lie.
- Effacer les noms des variables liées.

Si les formules deviennent identiques, alors elles sont α-équivalentes.

Exemple 4.4.8

Les deux formules $\forall x \exists y P(y, x)$ et $\forall y \exists x P(x, y)$ donnent :

$$\forall \; x \; \exists \; y \; P(\; y \; , \; x \;)$$

Savoir si deux formules sont α-équivalentes

Technique

- Tracer un trait entre chaque quantificateur et les variables qu'il lie.
- Effacer les noms des variables liées.

Si les formules deviennent identiques, alors elles sont α-équivalentes.

Exemple 4.4.8

Les deux formules $\forall x \exists y P(y, x)$ et $\forall y \exists x P(x, y)$ donnent :

$$\forall \; x \; \exists \; y \; P(\; y \; , \; x \;)$$

Savoir si deux formules sont α-équivalentes

Technique

- Tracer un trait entre chaque quantificateur et les variables qu'il lie.
- Effacer les noms des variables liées.

Si les formules deviennent identiques, alors elles sont α-équivalentes.

Exemple 4.4.8

Les deux formules $\forall x \exists y P(y, x)$ et $\forall y \exists x P(x, y)$ donnent :

$$\forall \quad \exists \quad P(\quad , \quad)$$

Exercice

Calculer la transformation pour

- $A = \forall x \forall y\ R(x, y, y)$
- $B = \forall x \forall y\ R(x, x, y)$

A et B sont-elles α-équivalentes ?

Preuve sans règle de copie

Dans l'environnement (i) $\exists x P(x)$:

Preuve sans règle de copie

Dans l'environnement (i) $\exists x P(x)$:

1	1	supposons $P(x)$

Preuve sans règle de copie

Dans l'environnement (i) $\exists x P(x)$:

1	1	supposons $P(x)$	
1	2	$\exists y P(y)$	$\exists$I 1, x

Preuve sans règle de copie

Dans l'environnement (i) $\exists x P(x)$:

1	1	supposons $P(x)$	
1	2	$\exists y P(y)$	$\exists$I 1, x
	3	donc $P(x) \Rightarrow \exists y P(y)$	$\Rightarrow$I 1, 2

Preuve sans règle de copie

Dans l'environnement $(i)\ \exists x P(x)$:

1	1	supposons $P(x)$	
1	2	$\exists y P(y)$	$\exists$I 1, x
	3	donc $P(x) \Rightarrow \exists y P(y)$	$\Rightarrow$I 1, 2
	4	$\exists y P(y)$	$\exists$E i, 3

Preuve sans règle de copie

Dans l'environnement $(i)\ \exists x P(x)$:

1	1	supposons $P(x)$	
1	2	$\exists y P(y)$	$\exists$I 1, x
	3	donc $P(x) \Rightarrow \exists y P(y)$	$\Rightarrow$I 1, 2
	4	$\exists y P(y)$	$\exists$E i, 3

Théorème (admis)

Soient A et A' deux formules fermées copies l'une de l'autre.
Alors il existe une preuve de A dans l'environnement A'.

Plan

Introduction

Règles et exemples

Règle de la copie

Les règles de l'égalité

Tactiques de preuves

Conclusion

Réflexivité et congruence

Deux règles caractérisent l'égalité :

- un terme est égal à lui-même
- si deux termes sont égaux, on peut les remplacer l'un par l'autre.

Réflexivité et congruence

Deux règles caractérisent l'égalité :

- un terme est égal à lui-même
- si deux termes sont égaux, on peut les remplacer l'un par l'autre.

$\dfrac{}{t{=}t}$	réflexivité	t est un terme
$\dfrac{s{=}t \qquad A{<}x{:=}s{>}}{A{<}x{:=}t{>}}$	congruence	s et t sont deux termes libres pour la variable x dans la formule A

Exemple 6.1.7

Prouvons que $s = t \Rightarrow t = s$ (symétrie)

Exemple 6.1.7

Prouvons que $s = t \Rightarrow t = s$ (symétrie)

1	1	supposons $s = t$

Exemple 6.1.7

Prouvons que $s = t \Rightarrow t = s$ (symétrie)

1	1	supposons $s = t$	
1	2	$s = s$	réflexivité

Exemple 6.1.7

Prouvons que $s = t \Rightarrow t = s$ (symétrie)

1	1	supposons $s = t$	
1	2	$s = s$	réflexivité
1	3	$t = s$	congruence 1, 2

$$\frac{s = t \quad (x = s) < x := s >}{(x = s) < x := t >}$$

Exemple 6.1.7

Prouvons que $s = t \Rightarrow t = s$ (symétrie)

1	1	supposons $s = t$	
1	2	$s = s$	réflexivité
1	3	$t = s$	congruence 1, 2
	3	donc $s = t \Rightarrow t = s$	$\Rightarrow$I 1, 3

$$\frac{s = t \quad (x = s) < x := s >}{(x = s) < x := t >}$$

Exemple 6.1.7

Prouvons que $s = t \Rightarrow t = s$ (symétrie)

1	1	supposons $s = t$	
1	2	$s = s$	réflexivité
1	3	$t = s$	congruence 1, 2
	3	donc $s = t \Rightarrow t = s$	$\Rightarrow$I 1, 3

$$\frac{s = t \quad (x = s) < x := s >}{(x = s) < x := t >}$$

Remarque : La variable x ne figure pas dans la preuve, elle sert à marquer l'endroit où on remplace s par t.

Dans les prochains exemples, on se contente de souligner cet endroit.

Exemple 6.1.8

Prouvons que $s = t \wedge t = u \Rightarrow s = u$ (transitivité)

Exemple 6.1.8

Prouvons que $s = t \wedge t = u \Rightarrow s = u$ (transitivité)

1	1	supposons $s = t \wedge t = u$

Exemple 6.1.8

Prouvons que $s = t \wedge t = u \Rightarrow s = u$ (transitivité)

1	1	supposons $s = t \wedge t = u$	
1	2	$s = t$	$\wedge$E1 1

Exemple 6.1.8

Prouvons que $s = t \wedge t = u \Rightarrow s = u$ (transitivité)

1	1	supposons $s = t \wedge t = u$	
1	2	$s = t$	$\wedge$E1 1
1	3	$t = u$	$\wedge$E2 1

Exemple 6.1.8

Prouvons que $s = t \wedge t = u \Rightarrow s = u$ (transitivité)

1	1	supposons $s = t \wedge t = u$	
1	2	$s = \underline{t}$	$\wedge$E1 1
1	3	$t = u$	$\wedge$E2 1
1	4	$s = \underline{u}$	congruence 3, 2

Exemple 6.1.8

Prouvons que $s = t \wedge t = u \Rightarrow s = u$ (transitivité)

1	1	supposons $s = t \wedge t = u$	
1	2	$s = \underline{t}$	$\wedge$E1 1
1	3	$t = u$	$\wedge$E2 1
1	4	$s = \underline{u}$	congruence 3, 2
	5	donc $s = t \wedge t = u \Rightarrow s = u$	$\Rightarrow$I 1, 4

Plan

Introduction

Règles et exemples

Règle de la copie

Les règles de l'égalité

Tactiques de preuves

Conclusion

Introduction

1. Deux tactiques de preuves qui correspondent à des formes de raisonnement mathématique :
   1.1 pour la règle $\exists E$, raisonner en avant avec une hypothèse d'existence,
   1.2 pour la règle $\forall I$, raisonner en arrière pour généraliser.

2. Application à un exemple.

Introduction

1. Deux tactiques de preuves qui correspondent à des formes de raisonnement mathématique :
   1.1 pour la règle $\exists E$, raisonner en avant avec une hypothèse d'existence,
   1.2 pour la règle $\forall I$, raisonner en arrière pour généraliser.

2. Application à un exemple.

Pas de tactique pour les règles $\forall E$ et $\exists I$ (ce sont celles qui rendent le système indécidable !)

Raisonner en avant avec une hypothèse d'existence

Soit Γ un ensemble de formules, x une variable, A et C des formules.

On cherche une preuve de C dans l'environnement $\Gamma, \exists x A$.

Raisonner en avant avec une hypothèse d'existence

Soit Γ un ensemble de formules, x une variable, A et C des formules.

On cherche une preuve de C dans l'environnement $\Gamma, \exists x A$.

Deux cas possibles :

- x n'est libre ni dans Γ, ni dans C.
- x est libre dans Γ ou C.

1er cas : x n'est libre ni dans Γ, ni dans C

Dans ce cas, la preuve peut toujours s'écrire :

supposons A	
$\boxed{\text{preuve de } C \text{ dans l'environnement } \Gamma, A}$	
donc $A \Rightarrow C$	$\Rightarrow$I 1,...
C	$\exists$E

$2^{\text{ème}}$ cas : x est libre dans Γ ou C

On choisit une variable y :

- « nouvelle », c'est-à-dire non libre dans Γ, C
- absente de A

puis on se ramène au cas précédent, via la règle de copie.

La preuve s'écrit alors :

$\exists y A < x := y >$	copie de $\exists x A$
supposons $A < x := y >$	
preuve de C dans l'environnement $\Gamma, A < x := y >$	
donc $A < x := y > \Rightarrow C$	$\Rightarrow$I 1,_
C	$\exists$E

Un exemple simple

Prouvons $\exists x P(x) \wedge \forall x \neg P(x) \Rightarrow \bot$.

Un exemple simple

Prouvons $\exists x P(x) \wedge \forall x \neg P(x) \Rightarrow \bot$.

1	1	supposons $\exists x P(x) \wedge \forall x \neg P(x)$
1	8	$\bot$
	9	donc $\exists x P(x) \wedge \forall x \neg P(x) \Rightarrow \bot$ $\quad \Rightarrow$I 1, 8

Un exemple simple

Prouvons $\exists x P(x) \wedge \forall x \neg P(x) \Rightarrow \bot$.

1	1	supposons $\exists x P(x) \wedge \forall x \neg P(x)$	
1	2	$\exists x P(x)$	$\wedge$E1 1
1	3	$\forall x \neg P(x)$	$\wedge$E1 1
1	8	$\bot$	
	9	donc $\exists x P(x) \wedge \forall x \neg P(x) \Rightarrow \bot$	$\Rightarrow$I 1, 8

Un exemple simple

Prouvons $\exists x P(x) \wedge \forall x \neg P(x) \Rightarrow \bot$.

1	1	supposons $\exists x P(x) \wedge \forall x \neg P(x)$	
1	2	$\exists x P(x)$	$\wedge$E1 1
1	3	$\forall x \neg P(x)$	$\wedge$E1 1
1,2	4	supposons $P(x)$	
1,2	6	$\bot$	
1	7	donc $P(x) \Rightarrow \bot$	
1	8	$\bot$	$\exists$E 2,7
	9	donc $\exists x P(x) \wedge \forall x \neg P(x) \Rightarrow \bot$	$\Rightarrow$I 1, 8

Un exemple simple

Prouvons $\exists x P(x) \wedge \forall x \neg P(x) \Rightarrow \bot$.

1	1	supposons $\exists x P(x) \wedge \forall x \neg P(x)$	
1	2	$\exists x P(x)$	$\wedge$E1 1
1	3	$\forall x \neg P(x)$	$\wedge$E1 1
1,2	4	supposons $P(x)$	
1,2	5	$\neg P(x)$	$\forall$E 3 x
1,2	6	$\bot$	$\Rightarrow$E 4,5
1	7	donc $P(x) \Rightarrow \bot$	
1	8	$\bot$	$\exists$E 2,7
	9	donc $\exists x P(x) \wedge \forall x \neg P(x) \Rightarrow \bot$	$\Rightarrow$I 1, 8

Remarques

La recherche de la preuve initiale a été réduite à la recherche d'une preuve de la même formule dans un environnement plus simple.

Remarques

La recherche de la preuve initiale a été réduite à la recherche d'une preuve de la même formule dans un environnement plus simple.

Mode de raisonnement appliqué en mathématiques quand on cherche une preuve d'une formule C avec l'hypothèse $\exists x P(x)$.

Remarques

La recherche de la preuve initiale a été réduite à la recherche d'une preuve de la même formule dans un environnement plus simple.

Mode de raisonnement appliqué en mathématiques quand on cherche une preuve d'une formule C avec l'hypothèse $\exists x P(x)$.

Cela revient à introduire une constante « nouvelle » a vérifiant $P(a)$ et à prouver C sous l'hypothèse $P(a)$.

Raisonner en arrière pour généraliser

On cherche une preuve de $\forall x A$ dans l'environnement Γ.

Raisonner en arrière pour généraliser

On cherche une preuve de $\forall x A$ dans l'environnement Γ.

Deux cas possibles :

- x n'est pas libre dans Γ.
- x est libre dans Γ.

1er cas : x n'est pas libre dans Γ

$$\boxed{\text{preuve de } A \text{ dans l'environnement } \Gamma}$$
$$\forall x A \quad \forall \mathrm{I}$$

$2^{\text{ème}}$ cas : x est libre dans Γ

On choisit une variable y :

- « nouvelle », c'est-à-dire non libre dans Γ
- absente de A

puis on se ramène au cas précédent, via la règle de copie.

La preuve s'écrit alors :

preuve de $A < x := y >$ dans l'environnement Γ	
$\forall y A < x := y >$	$\forall$I
$\forall x A$	copie de la formule précédente

Un exemple simple

Prouvons $\forall x P(x) \Rightarrow \forall y P(y)$ sans copie.

Un exemple simple

Prouvons $\forall x P(x) \Rightarrow \forall y P(y)$ sans copie.

1	1	supposons $\forall x P(x)$
1	3	$\forall y P(y)$
	4	donc $\forall x P(x) \Rightarrow \forall y P(y)$ $\Rightarrow$I 1, 4

Un exemple simple

Prouvons $\forall x P(x) \Rightarrow \forall y P(y)$ sans copie.

1	1	supposons $\forall x P(x)$	
		$P(y)$	
1	3	$\forall y P(y)$	$\forall$I 2
	4	donc $\forall x P(x) \Rightarrow \forall y P(y)$	$\Rightarrow$I 1, 4

Un exemple simple

Prouvons $\forall x P(x) \Rightarrow \forall y P(y)$ sans copie.

1	1	supposons $\forall x P(x)$	
1	2	$P(y)$	$\forall$E 1 y
1	3	$\forall y P(y)$	$\forall$I 2
	4	donc $\forall x P(x) \Rightarrow \forall y P(y)$	$\Rightarrow$I 1, 4

Remarque

La recherche de la preuve initiale a été réduite à la recherche d'une preuve d'une formule plus simple dans le même environnement.

Remarque

La recherche de la preuve initiale a été réduite à la recherche d'une preuve d'une formule plus simple dans le même environnement.

Mode de raisonnement appliqué en mathématiques quand on cherche une preuve de $\forall x P(x)$.

Remarque

La recherche de la preuve initiale a été réduite à la recherche d'une preuve d'une formule plus simple dans le même environnement.

Mode de raisonnement appliqué en mathématiques quand on cherche une preuve de $\forall x P(x)$.

Cela revient à introduire une variable « nouvelle » y et à prouver la formule $P(y)$.
Puis on conclut : puisque le choix de y est arbitraire, on a $\forall x P(x)$.

Un exemple d'application des tactiques

On définit « il existe un et un seul x » (noté en bref $\exists! x$) par :

- $\exists! x P(x) \doteq \exists x (P(x) \wedge \forall y (P(y) \Rightarrow x = y))$

Un exemple d'application des tactiques

On définit « il existe un et un seul x » (noté en bref $\exists! x$) par :

- $\exists! x P(x) \doteq \exists x (P(x) \wedge \forall y (P(y) \Rightarrow x = y))$

En séparant l'existence de x et son unicité, on peut aussi la définir par :

- $\exists! x P(x) \doteq \exists x P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y)$.

Ces deux définitions sont bien sûr équivalentes : on montre ici formellement que **la première implique la deuxième**.

Un exemple d'application des tactiques

On définit « il existe un et un seul x » (noté en bref $\exists! x$) par :

- $\exists! x P(x) \doteq \exists x (P(x) \wedge \forall y (P(y) \Rightarrow x = y))$

En séparant l'existence de x et son unicité, on peut aussi la définir par :

- $\exists! x P(x) \doteq \exists x P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y)$.

Ces deux définitions sont bien sûr équivalentes : on montre ici formellement que **la première implique la deuxième**.

Comme la preuve est longue, nous allons la décomposer.

6.2.3 Plan de la preuve

$$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y)) \Rightarrow \exists x P(x) \wedge \forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$$

On applique les deux tactiques suivantes :

- Pour prouver $A \Rightarrow B$, supposer A et déduire B
- Pour prouver $B_1 \wedge B_2$, prouver B_1 et prouver B_2.

6.2.3 Plan de la preuve

$$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y)) \Rightarrow \exists x P(x) \wedge \forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$$

On applique les deux tactiques suivantes :

- Pour prouver $A \Rightarrow B$, supposer A et déduire B
- Pour prouver $B_1 \wedge B_2$, prouver B_1 et prouver B_2.

1	supposons $\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	
1	preuve de $\exists x P(x)$ dans l'environnement 1	
1	preuve de $\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$ dans l'environnement 1	
1	$\exists x P(x) \wedge \forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\wedge$I
	donc $\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y)) \Rightarrow \exists x P(x) \wedge \forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\Rightarrow$I

6.2.3 Application de la tactique utilisant une hypothèse d'existence

Preuve de $\exists x P(x)$ dans l'environnement
$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$

6.2.3 Application de la tactique utilisant une hypothèse d'existence

Preuve de $\exists x P(x)$ dans l'environnement
$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$

contexte	N°	formule	règle
	i	$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	
1	1	supposons $P(x) \wedge \forall y(P(y) \Rightarrow x = y)$	
1	2	$P(x)$	$\wedge$E1 1
1	3	$\exists x P(x)$	$\exists$I 2, x
	4	donc $P(x) \wedge \forall y(P(y) \Rightarrow x = y) \Rightarrow \exists x P(x)$	$\Rightarrow$I 1,3
	5	$\exists x P(x)$	$\exists$E i, 4

6.2.3 Application de la tactique pour obtenir une conclusion générale : plan de preuve

Preuve de $\forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y)$
dans l'environnement $\exists x (P(x) \wedge \forall y (P(y) \Rightarrow x = y))$

On applique les tactiques suivantes :

1. « raisonner en avant en utilisant une hypothèse existentielle »
2. « raisonner en arrière pour obtenir une conclusion générale » (2 fois)
3. Pour prouver $A \Rightarrow B$, supposer A et déduire B

6.2.3 Application de la tactique pour obtenir une conclusion générale : preuve

contexte	N°	formule	règle
	i	$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	

6.2.3 Application de la tactique pour obtenir une conclusion générale : preuve

contexte	N°	formule	règle
	i	$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	
1	1	supposons $P(x) \wedge \forall y(P(y) \Rightarrow x = y)$	
1	14	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	
	15	donc $(P(x) \wedge \forall y(P(y) \Rightarrow x = y)) \Rightarrow \forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\Rightarrow$I 1, 14
	16	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\exists$E i, 15

6.2.3 Application de la tactique pour obtenir une conclusion générale : preuve

contexte	N°	formule	règle
	i	$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	
1	1	supposons $P(x) \wedge \forall y(P(y) \Rightarrow x = y)$	
1	13	$\forall u \forall y(P(u) \wedge P(y) \Rightarrow u = y)$	$\forall$I 12
1	14	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	copie de 13
	15	donc $(P(x) \wedge \forall y(P(y) \Rightarrow x = y)) \Rightarrow \forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\Rightarrow$I 1, 14
	16	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\exists$E i, 15

6.2.3 Application de la tactique pour obtenir une conclusion générale : preuve

contexte	N°	formule	règle
	i	$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	
1	1	supposons $P(x) \wedge \forall y(P(y) \Rightarrow x = y)$	
1	11	$P(u) \wedge P(y) \Rightarrow u = y$	
1	12	$\forall y(P(u) \wedge P(y) \Rightarrow u = y)$	$\forall$I 11
1	13	$\forall u \forall y(P(u) \wedge P(y) \Rightarrow u = y)$	$\forall$I 12
1	14	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	copie de 13
	15	donc $(P(x) \wedge \forall y(P(y) \Rightarrow x = y)) \Rightarrow \forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\Rightarrow$I 1, 14
	16	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\exists$E i, 15

6.2.3 Application de la tactique pour obtenir une conclusion générale : preuve

contexte	N°	formule	règle
	i	$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	
1	1	supposons $P(x) \wedge \forall y(P(y) \Rightarrow x = y)$	
1,2	2	supposons $P(u) \wedge P(y)$	
1,2	10	$u = y$	
1	11	donc $P(u) \wedge P(y) \Rightarrow u = y$	$\Rightarrow$I 2, 10
1	12	$\forall y(P(u) \wedge P(y) \Rightarrow u = y)$	$\forall$I 11
1	13	$\forall u \forall y(P(u) \wedge P(y) \Rightarrow u = y)$	$\forall$I 12
1	14	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	copie de 13
	15	donc $(P(x) \wedge \forall y(P(y) \Rightarrow x = y)) \Rightarrow \forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\Rightarrow$I 1, 14
	16	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\exists$E i, 15

6.2.3 Application de la tactique pour obtenir une conclusion générale : preuve

contexte	N°	formule	règle
	i	$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	
1	1	supposons $P(x) \wedge \forall y(P(y) \Rightarrow x = y)$	
1,2	2	supposons $P(u) \wedge P(y)$	
1,2	3	$\forall y(P(y) \Rightarrow x = y)$	$\wedge$E2 1
1,2	10	$u = y$	
1	11	donc $P(u) \wedge P(y) \Rightarrow u = y$	$\Rightarrow$I 2, 10
1	12	$\forall y(P(u) \wedge P(y) \Rightarrow u = y)$	$\forall$I 11
1	13	$\forall u \forall y(P(u) \wedge P(y) \Rightarrow u = y)$	$\forall$I 12
1	14	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	copie de 13
	15	donc $(P(x) \wedge \forall y(P(y) \Rightarrow x = y)) \Rightarrow \forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\Rightarrow$I 1, 14
	16	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\exists$E i, 15

6.2.3 Application de la tactique pour obtenir une conclusion générale : preuve

contexte	N°	formule	règle
	i	$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	
1	1	supposons $P(x) \wedge \forall y(P(y) \Rightarrow x = y)$	
1,2	2	supposons $P(u) \wedge P(y)$	
1,2	3	$\forall y(P(y) \Rightarrow x = y)$	$\wedge$E2 1
1,2	4	$P(u)$	$\wedge$E1 2
1,2	5	$P(u) \Rightarrow x = u$	$\forall$E 3, u
1,2	6	$x = u$	$\Rightarrow$E 4, 5
1,2	10	$u = y$	
1	11	donc $P(u) \wedge P(y) \Rightarrow u = y$	$\Rightarrow$I 2, 10
1	12	$\forall y(P(u) \wedge P(y) \Rightarrow u = y)$	$\forall$I 11
1	13	$\forall u \forall y(P(u) \wedge P(y) \Rightarrow u = y)$	$\forall$I 12
1	14	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	copie de 13
	15	donc $(P(x) \wedge \forall y(P(y) \Rightarrow x = y)) \Rightarrow \forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\Rightarrow$I 1, 14
	16	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\exists$E i, 15

6.2.3 Application de la tactique pour obtenir une conclusion générale : preuve

contexte	N°	formule	règle
	i	$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	
1	1	supposons $P(x) \wedge \forall y(P(y) \Rightarrow x = y)$	
1,2	2	supposons $P(u) \wedge P(y)$	
1,2	3	$\forall y(P(y) \Rightarrow x = y)$	$\wedge$E2 1
1,2	4	$P(u)$	$\wedge$E1 2
1,2	5	$P(u) \Rightarrow x = u$	$\forall$E 3, u
1,2	6	$x = u$	$\Rightarrow$E 4, 5
1,2	7	$P(y)$	$\wedge$E2 2
1,2	8	$P(y) \Rightarrow x = y$	$\forall$E 3, y
1,2	9	$x = y$	$\Rightarrow$E 7, 8
1,2	10	$u = y$	
1	11	donc $P(u) \wedge P(y) \Rightarrow u = y$	$\Rightarrow$I 2, 10
1	12	$\forall y(P(u) \wedge P(y) \Rightarrow u = y)$	$\forall$I 11
1	13	$\forall u \forall y(P(u) \wedge P(y) \Rightarrow u = y)$	$\forall$I 12
1	14	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	copie de 13
	15	donc $(P(x) \wedge \forall y(P(y) \Rightarrow x = y)) \Rightarrow \forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\Rightarrow$I 1, 14
	16	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\exists$E i, 15

6.2.3 Application de la tactique pour obtenir une conclusion générale : preuve

contexte	N°	formule	règle
	i	$\exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$	
1	1	supposons $P(x) \wedge \forall y(P(y) \Rightarrow x = y)$	
1,2	2	supposons $P(u) \wedge P(y)$	
1,2	3	$\forall y(P(y) \Rightarrow x = y)$	$\wedge$E2 1
1,2	4	$P(u)$	$\wedge$E1 2
1,2	5	$P(u) \Rightarrow x = u$	$\forall$E 3, u
1,2	6	$x = u$	$\Rightarrow$E 4, 5
1,2	7	$P(y)$	$\wedge$E2 2
1,2	8	$P(y) \Rightarrow x = y$	$\forall$E 3, y
1,2	9	$\underline{x} = y$	$\Rightarrow$E 7, 8
1,2	10	$\underline{u} = y$	congruence 6, 9
1	11	donc $P(u) \wedge P(y) \Rightarrow u = y$	$\Rightarrow$I 2, 10
1	12	$\forall y(P(u) \wedge P(y) \Rightarrow u = y)$	$\forall$I 11
1	13	$\forall u \forall y(P(u) \wedge P(y) \Rightarrow u = y)$	$\forall$I 12
1	14	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	copie de 13
	15	donc $(P(x) \wedge \forall y(P(y) \Rightarrow x = y)) \Rightarrow \forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\Rightarrow$I 1, 14
	16	$\forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y)$	$\exists$E i, 15

Conclusion

Toute la difficulté des preuves est donc concentrée dans les règles $\forall$E et $\exists$I :

- dans le raisonnement en avant, il faut trouver les bonnes instanciations des formules commençant par $\forall$
- dans le raisonnement en arrière, il faut trouver la bonne instance permettant de déduire une formule commençant par $\exists$

Les théorèmes d'incomplétude de Kurt Gödel (1906-1978)

Premier théorème d'incomplétude (1931)

Tout système logique qui permet de formaliser l'arithmétique permet également d'énoncer :

« *Cet énoncé est indémontrable* ».

- ou bien cet énoncé est faux ; il est donc aussi démontrable, et notre système est incohérent
- ou bien cet énoncé est vrai ; il est donc indémontrable, et notre système est incomplet

Les théorèmes d'incomplétude de Kurt Gödel (1906-1978)

Premier théorème d'incomplétude (1931)

Tout système logique qui permet de formaliser l'arithmétique permet également d'énoncer :

« *Cet énoncé est indémontrable* ».

- ou bien cet énoncé est faux ; il est donc aussi démontrable, et notre système est incohérent
- ou bien cet énoncé est vrai ; il est donc indémontrable, et notre système est incomplet

Second théorème d'incomplétude

Aucun système logique ne peut démontrer sa propre cohérence.

Plan

Introduction

Règles et exemples

Règle de la copie

Les règles de l'égalité

Tactiques de preuves

Conclusion

Aujourd'hui

- Déduction Naturelle au premier ordre
  - Règles d'introduction et élimination pour les quantificateurs
  - Copie, égalité en déduction naturelle
- Tactiques
- Cohérence et complétude du système (admises)

Plan du Semestre

- Logique propositionnelle
- Résolution propositionnelle
- Déduction naturelle propositionnelle

PARTIEL

- Logique du premier ordre
- Base de la démonstration automatique (“résolution au premier ordre”)
- Déduction naturelle au premier ordre

EXAMEN

Partiel INF402

Frédéric Prost Benjamin Wack

16 mars 2023
2 pages
Total : 120 points
Durée : 2h00

Documents autorisés : une feuille recto verso de notes manuscrites format A4.
Le barème est *indicatif*, les points correspondent au nombre de minutes nécessaires pour réaliser les exercices. L'épreuve sera notée sur 120 points.
Le résultat d'une question peut être admis pour s'en servir dans la suite de l'énoncé. Les exercices peuvent être traités dans l'ordre de votre choix à condition de les numéroter clairement.

Exercice 1 (Formalisation (20 points))

Considérons le raisonnement suivant.

- *Dieu est omnibienveillant.*
- *Si la causalité était transitive, on en déduirait que : si Dieu est la cause de l'existence de l'humanité, il s'ensuit que Dieu a causé le changement climatique si c'est l'existence de l'humanité qui l'a causé.*
- *Or, tout bon scientifique doit admettre que c'est l'activité humaine qui a causé les problèmes de climat actuels.*
- *Mais bien que tout bon croyant sache que Dieu a créé l'humanité, un Dieu responsable du changement climatique global ne peut pas être omnibienveillant.*
- *On en déduit que la causalité est non-transitive.*

1. *Formalisez ce raisonnement en logique propositionnelle, en reformulant les affirmations si nécessaire. Ne perdez pas de temps à comprendre les affirmations, concentrez-vous sur les connecteurs logiques.*
2. *Trouvez une assignation qui satisfait les hypothèses.*
3. *Que pouvez-vous en conclure à propos de ce raisonnement ?*

Réponse:

On définit les variables propositionnelles :

P1 :	Dieu est omnibénévolent.
P2 :	Dieu a causé l'existence de l'humanité.
P3 :	Dieu a causé le changement climatique.
Q :	La causalité est transitive.
R :	L'humanité est la cause du changement climatique.

Le raisonnement se traduit par les formules :

$$\begin{array}{l} P1 \\ Q \implies (P2 \implies (R \implies P3)) \\ R \\ P2 \wedge (P3 \implies \neg P1) \\ \models \neg Q \end{array}$$

L'assignation suivante :

P1 :	$\top$
P2 :	$\top$
P3 :	$\bot$
Q :	$\bot$
R :	$\top$.

est le seul modèle des hypothèses.
Comme c'est aussi un modèle de $\neg Q$, le raisonnement est correct.

□

Exercice 2 (Tables de vérité et formes normales (20 points))
Soit $A = \neg x \Rightarrow y \vee z \Rightarrow \neg(z \wedge y) \Rightarrow x$.

1. *Réécrivez cette formule sous forme d'arbre.* *(3 points)*
2. *Construisez la table de vérité de A.* *(5 points)*
3. *Déterminez une forme normale disjonctive de A et simplifiez-la le plus possible.* *(5 points)*
4. *Déterminez une forme normale conjonctive de A et simplifiez-la le plus possible.* *(5 points)*
5. *Donnez un modèle et un contre-modèle de A.* *(2 points)*

Réponse:

1. La formule stricte est $A = \Big(\neg x \Rightarrow \big((y \vee z) \Rightarrow (\neg(z \wedge y) \Rightarrow x)\big)\Big)$.
   qui peut ensuite facilement être écrite sous forme d'arbre.
2. La table de vérité est :

x	y	z	$\neg x$	$y \vee z$	$\neg(y \wedge z)$	A
0	0	0	1	0	1	1
0	0	1	1	1	1	0
0	1	0	1	1	1	0
0	1	1	1	1	0	1
1	0	0	0	0	1	1
1	0	1	0	1	1	1
1	1	0	0	1	1	1
1	1	1	0	1	0	1

3. La FND est $x + \bar{y}.\bar{z} + y.z$
4. La FNC est $(x + y + \bar{z}).(x + \bar{y} + z)$
5. Des modèles et contre-modèles se lisent facilement dans la table de vérité et/ou sur les formes normales.

□

Exercice 3 (Résolution (25 points))

Soient

$$F = \neg(\neg p \wedge \neg q \wedge \neg r) \wedge (\neg p \vee q \vee r)$$

et

$$G = q \vee r$$

Dans tout cet exercice, les seules équivalences remarquables autorisées sont la distributivité et les lois de De Morgan. En particulier, il est interdit d'utiliser les règles de simplification.

1. *Transformez F et G en produit de clauses.*
   Si nécessaire dans la suite de l'exercice, vous pourrez également calculer les clauses pour $\neg F$ et/ou $\neg G$.
2. *Parmi les propriétés suivantes, dites lesquelles sont vérifiées, et justifiez vos réponses en utilisant la résolution et les lois de De Morgan :*
   (a) $F \models G$
   (b) $G \models F$
   (c) $F \equiv G$
   (d) $\neg F \vee G$ *est une tautologie.*
   (e) $\neg F \vee G$ *est une contradiction.*

Réponse: Considérons les clauses correspondant aux formules F et G :

$$F_c = \{(p + q + r), (\overline{p} + q + r)\}$$

$$G_c = \{q + r\}$$

1. $F \models G$ si et seulement si $F \cup \neg G$ est insatisfaisable, ie permet de démontrer l'absurde par résolution.

$$F \cup \neg G = \{(p + q + r), (\overline{p} + q + r), \neg q, \neg r\}$$

1.	$p + q + r$	*Hypothse*
2.	$\overline{p} + q + r$	*Hypothse*
3.	$q + r$	*Resolvant de* 1.2.
4.	$\neg q$	*Hypothse*
5.	r	*Resolvant de* 3.4.
6.	$\neg r$	*Hypothse*
7.	$\perp$	*Resolvant de* 5.6.

2. Comme à la question précédente avec $G \cup \neg F$

$$G_c \cup \neg F_c = \{q + r, p + \overline{q}, p + \overline{r}, \overline{q} + \overline{p}, \overline{q}, \overline{q} + \overline{r}, \overline{p} + \overline{r}, \overline{q} + \overline{r}, \overline{r}\}$$

1.	$q + r$	*Hypothse*
2.	$\overline{r}$	*Hypothse*
3.	q	*Resolvant de* 1.2.
4.	$\overline{q}$	*Hypothse*
5.	$\perp$	*Resolvant de* 3.4.

3. $F \equiv G$ puisque $F \models G$ et $G \models F$.
4. $\neg F \vee G$ est une tautologie puisque $F \equiv G$ (loi du tiers exclus).
5. $\neg F \vee G$ n'est pas une contradiction puisque c'est une tautologie.

□

Exercice 4 (DPLL (20 points))

Déterminez **en appliquant l'algorithme DPLL** *si les ensembles de clauses suivants sont satisfaisables. Le cas échéant, vous donnerez le modèle exhibé par DPLL.*
Vous donnerez une trace complète du déroulement de l'algorithme étiquetée par les règles utilisées.

1. $\{a+b+\overline{c},\ \overline{a}+c+d,\ e+a+b,\ \overline{e}+c+e,\ b+\overline{c},\ \overline{b}+d,\ \overline{a}+\overline{c},\ c+\overline{d}\}$ *(10 points)*
2. $\{a+d,\ a+\overline{b}+\overline{c}+\overline{d},\ a+b+c+\overline{d},\ \overline{a}+\overline{b}+\overline{c}+d,\ \overline{a}+c,\ \overline{b}+a+c,\ \overline{c}+b,\ \overline{d}+\overline{a}\}$ *(10 points)*

Réponse:

1. $\{a+b+\overline{c},\ \overline{a}+c+d,\ e+a+b,\ \overline{e}+c+e,\ b+\overline{c},\ \overline{b}+d,\ \overline{a}+\overline{c},\ c+\overline{d}\}$
   RED : $\{\overline{a}+c+d,\ e+a+b,\ b+\overline{c},\ \overline{b}+d,\ \overline{a}+\overline{c},\ c+\overline{d}\}$
   ELI $e=1$: $\{\overline{a}+c+d,\ b+\overline{c},\ \overline{b}+d,\ \overline{a}+\overline{c},\ c+\overline{d}\}$
   ELI $a=0$: $\{b+\overline{c},\ \overline{b}+d,\ c+\overline{d}\}$
   Branche $b=0$: $\{\overline{c},\ c+\overline{d}\}$
   ELI $d=0$: $\{\overline{c}\}$
   ELI $c=0$: $\{\}$
   On a un modèle $a=0, b=0, c=0, d=0, e=1$
2. $\{a+d,\ a+\overline{b}+\overline{c}+\overline{d},\ a+b+c+\overline{d},\ \overline{a}+\overline{b}+\overline{c}+d,\ \overline{a}+c,\ \overline{b}+a+c,\ \overline{c}+b,\ \overline{d}+\overline{a}\}$
   $\{a+d,\ a+\overline{b}+\overline{c}+\overline{d},\ a+b+c+\overline{d},\ \overline{a}+\overline{b}+\overline{c}+d,\ \overline{a}+c,\ \overline{b}+a+c,\ \overline{c}+b,\ \overline{d}+\overline{a}\}$
   — Branche $a=0$: $\{d,\ \overline{b}+\overline{c}+\overline{d},\ b+c+\overline{d},\ \overline{b}+c,\ \overline{c}+b\}$
     RU $d=1$: $\{\overline{b}+\overline{c},\ b+c,\ \overline{b}+c,\ \overline{c}+b\}$
     — Branche $b=0$: $\{c,\ \overline{c}\}$
       RU : $\perp$
     — Branche $b=1$: $\{\overline{c},\ c\}$
       RU : $\perp$
   — Branche $a=1$: $\{\overline{b}+\overline{c}+d,\ c,\ \overline{c}+b,\ \overline{d}\}$
     RU $c=1, d=0$: $\{\overline{b},\ b\}$
     RU : $\perp$
   L'ensemble de clauses est insatisfaisable.

□

Exercice 5 (Déduction naturelle (20 points))
Prouver, **par déduction naturelle**, *les formules suivantes.*

Chaque formule de la preuve devra être justifiée ; il vous est recommandé de présenter vos preuves sous forme de tableaux comme vu en cours.

1. $A \ = \ (p \vee q \Rightarrow s \wedge t) \Rightarrow (q \Rightarrow s)$ *(10 points)*
2. $B \ = \ (p \vee \neg q) \Rightarrow \big((p \Rightarrow \neg q) \Rightarrow (q \Rightarrow r)\big)$ *(10 points)*

Réponse:

1. $A \ = \ (p \vee q \Rightarrow s \wedge t) \Rightarrow (q \Rightarrow s)$

```
1   assume p + q => s & t.
2     assume q.
3       p + q.                                  +I2 2
4       s & t.                                  =>E 1,3
5       s.                                      &E1 4
6     therefore q => s.                         =>I 2,5
7   therefore (p + q => s & t) => q => s.       =>I 1,6
```

2. $B \ = \ (p \vee \neg q) \Rightarrow \big((p \Rightarrow \neg q) \Rightarrow (q \Rightarrow r)\big)$

```
1   assume p + -q.
2     assume p => -q.
3       assume q.
4         assume p.
5           -q.                                 =>E 2,4
6           F.                                  =>E 5,3
7           r.                                  Efq 6
8         therefore p => r.                     =>I 4,7
9         assume -q.
```

```
10          F.                                          =>E 9,3
11          r.                                          Efq 10
12        therefore -q => r.                            =>I 9,11
13        r.                                            +ES 1,8,12
14      therefore q => r.                               =>I 3,13
15    therefore (p => -q) => q => r.                    =>I 2,14
16  therefore p + -q => (p => -q) => q => r.            =>I 1,15
```

□

Exercice 6 (Démonstration par récurrence, exo du poly (15 points))

Démontrez que toute formule construite avec une seule variable p, uniquement les connecteurs binaires $\vee$ et $\wedge$ et sans négation est équivalente à une formule de taille 0.

Si vous trouvez des cas très semblables, vous pouvez raccourcir la démonstration en admettant une partie d'entre eux.

Réponse: Nous effectuons une preuve par récurrence sur la taille des formules. Considérons une formule F construite avec une seule variable p, uniquement $\vee$ et $\wedge$ et sans négation.

Cas de base : $|F| = 0$. La récurrence est trivialement vérifiée dans ce cas.

Induction : $|F| = n + 1$ et supposons que toutes les formules construite avec une seule variable p, uniquement $\vee$ et $\wedge$ et sans négation de taille inférieure ou égale à n sont équivalentes à une formule de taille 0. Nous avons deux cas duaux, où $|A| \leq n$ et $|B| \leq n$, donc par hypothèse de récurrence A et B sont donc équivalentes à une formule de taille 0, c'est-à-dire 0, 1, ou p :

- $F = (A \vee B)$. Nous avons donc 9 cas à examiner :
  - $A = 0$
    - $B = 0$, donc $F = 0$.
    - $B = 1$, donc $F = 1$.
    - $B = p$, donc $F = p$.
  - $A = 1$
    - $B = 0$, donc $F = 1$.
    - $B = 1$, donc $F = 1$.
    - $B = p$, donc $F = 1$.
  - $A = p$
    - $B = 0$, donc $F = p$.
    - $B = 1$, donc $F = 1$.
    - $B = p$, donc $F = p$.
- $F = (A \wedge B)$. Nous avons donc 9 cas à examiner :
  - $A = 0$
    - $B = 0$, donc $F = 0$.
    - $B = 1$, donc $F = 0$.
    - $B = p$, donc $F = 0$.
  - $A = 1$
    - $B = 0$, donc $F = 0$.
    - $B = 1$, donc $F = 1$.
    - $B = p$, donc $F = p$.
  - $A = p$
    - $B = 0$, donc $F = 0$.
    - $B = 1$, donc $F = p$.
    - $B = p$, donc $F = p$.

□

Examen INF402

Frédéric Prost Benjamin Wack

16 mai 2023
2 pages
Total : 120 points (noté sur 100)
Durée : 2h00

Documents autorisés : une feuille recto verso de notes manuscrites format A4.
Le barème est *indicatif*, les points correspondent au nombre de minutes nécessaires pour réaliser les exercices. **L'épreuve sera notée sur 100 points (total divisé par 5), il est donc possible de faire l'impasse sur un des exercices sans pénalité.**
Le résultat d'une question peut être admis pour s'en servir dans la suite de l'énoncé. Les exercices peuvent être traités dans l'ordre de votre choix à condition de les numéroter clairement.

Exercice 1 (Formalisation (20 points))**.**
Nous considérons la signature $\Sigma = \{animal^{r1},\ plante^{r1},\ mange^{r2},\ pluspetit^{r2}\}$.

1. (12 points) *Exprimer en logique du premier ordre et en utilisant la signature* Σ *les phrases suivantes :*
   (a) *Il existe des animaux qui mangent à la fois des plantes et des animaux.*
   (b) *Aucune plante ne mange d'animal.*
   (c) *Tous les animaux mangent soit toutes les plantes, soit certains animaux plus petits qu'eux.*
   (d) *Les animaux qui mangent d'autres animaux mangent tous les animaux plus petits qu'eux.*
2. (8 points) *Exprimer en français les propositions logiques suivantes :*
   (a) $\forall x \neg(animal(x) \Leftrightarrow plante(x))$
   (b) $\exists x(plante(x) \wedge \forall y(plante(y) \wedge x \neq y \Rightarrow pluspetit(y,x))$
   (c) $\exists x \exists y(animal(x) \wedge animal(y) \wedge mange(x,y) \wedge \forall z(mange(x,z) \Rightarrow y = z))$

□

Exercice 2 (Expansions (20 points))**.**
À l'aide de la méthode des expansions, déterminer ***un contre-modèle*** *de chacune de ces formules :*

1. (10 points) $A = \big(\forall x Q(x) \implies \forall x P(x)\big) \implies \forall x\big(Q(x) \implies P(x)\big)$.
2. (10 points) $B = \forall x \exists y\big((x = y) \implies P(x,y)\big) \implies \forall x \forall y\big(P(x,y) \implies (x = y)\big)$.

□

Exercice 3 (Unification (15 points))**.**
Pour chacune des équations suivantes, déterminez si elle admet une solution, et si oui déterminez un unificateur le plus général. Vous utiliserez pour cela l'algorithme du cours dont vous détaillerez les étapes en donnant à chaque fois le nom des règles utilisées.
a et b sont des constantes ; f, g et h sont des fonctions.

1. (5 points) $f\big(x, f(u,x)\big) = f\big(f(y,a), f(z, f(b,z))\big)$
2. (5 points) $h\big(z, f(y, g(a,x))\big) = h\big(f(b,b), f(z, g(x, h(x,x)))\big)$
3. (5 points) $f\big(h(g(a,x), g(z,z)), g(z,z)\big) = f\big(h(y,y), y\big)$

□

Exercice 4 (Skolémisation (15 points)).

Soit A la formule suivante : $A = \Big(\exists x \forall y Q(x,y) \wedge \forall x\big(Q(x,x) \Rightarrow \exists y R(y,x)\big)\Big) \Rightarrow \exists y \exists x R(x,y)$

Montrez que A est valide en montrant que $\neg A$ est contradictoire par la méthode de Herbrand, en suivant les étapes suivantes :

1. (10 points) *Skolémisez $\neg A$ puis mettez-la en forme clausale.*
2. (5 points) *Proposez des instances contradictoires des clauses obtenues et montrez la contradiction par résolution propositionnelle.*

□

Exercice 5 (Résolution (10 points)).

Considérons l'ensemble suivant de clauses du premier ordre :

$$\Gamma = \{\overline{P(x)} + \overline{P(f(a))} + Q(y), P(y), \overline{P(g(b,x))} + \overline{Q(b)}, \overline{P(f(x))} + Q(x)\}$$

Nous voulons montrer que cet ensemble de clauses est insatisfaisable par instanciation ou par résolution. ***Au choix :*** *(vous n'obtiendrez pas le double de points en utilisant les deux méthodes)*

1. (10 points) *Trouvez des instances contradictoires de ces clauses et montrez par résolution propositionnelle que ces instances sont contradictoires.*

   OU

2. (10 points) *Donnez une preuve directe du caractère contradictoire de Γ par factorisation, copie et résolution binaire. Il est possible que toutes les règles ne soient pas utilisées.*

□

Exercice 6 (Déduction naturelle (20 points)).

Démontrez les formules suivantes en déduction naturelle au premier ordre. Comme vu en cours et en TD, on numérotera les lignes de la preuve, et on indiquera pour chaque ligne la formule démontrée, le contexte, la règle appliquée et les prémisses utilisées.

1. $\forall y \exists x \neg Q(x,y) \Rightarrow \neg \forall x \forall y Q(x,y)$
2. $\forall x \forall y (P(x) \wedge \neg P(y) \Rightarrow x \neq y)$

□

Exercice 7 (Ensemble complet et incomplet de connecteurs, exercice du poly (20 points)).

Un ensemble de constantes et de connecteurs est dit complet*, si toute fonction booléenne est exprimable avec ces constantes et ces connecteurs. Nous avons notamment vu en cours un théorème qui exprime que l'ensemble $\{\bot, \top, \neg, \vee, \wedge\}$ est complet.*

1. *Soit $\mid$ l'opération suivante : $x \mid y$ est vrai si et seulement si ni l'un, ni l'autre ne sont vrais, autrement dit $x \mid y = 1$ si et seulement si $x = 0$ et $y = 0$. Cette opération est aussi appelée* ni *car $x \mid y$ est vrai si et seulement si ni x, ni y ne sont vrais.*

   Montrer que cette opération (seule !) est complète.

2. *Montrer que l'ensemble $\{\top, \Rightarrow\}$ est incomplet.*

   Indication *: on pourra montrer que toute formule construite uniquement avec $\top$ et $\Rightarrow$ prend la valeur 1 quand toutes ses variables prennent la valeur 1. On en déduira alors assez facilement une fonction booléenne qui ne peut pas être exprimée avec cet ensemble.*

□

Corrigés

Exercice 1, page 1

1. $\exists x \exists y \exists z (animal(x) \wedge plante(y) \wedge animal(z) \wedge mange(x,y) \wedge mange(x,z))$
2. $\forall x (plante(x) \Rightarrow \forall y (animal(y) \Rightarrow \neg mange(x,y)))$
3. $\forall x (animal(x) \Rightarrow \forall y (plante(y) \Rightarrow mange(x,y)) \vee \exists y (animal(y) \wedge pluspetit(y,x) \wedge mange(x,y)))$
4. $\forall x (animal(x) \wedge \exists y (animal(y) \wedge mange(x,y)) \Rightarrow \forall z (animal(z) \wedge pluspetit(z,x) \Rightarrow mange(x,z))$
5. Tout être est soit une plante soit un animal, mais jamais les deux à la fois.
6. Il existe une plante plus grande que toutes les autres.
7. Il existe un animal qui ne mange qu'une seule espèce d'animal.

Exercice 2, page 1

1. La 1-expansion est valide : $(Q(0) \implies P(0)) \implies (Q(0) \implies P(0))$
   La 2-expansion est : $(Q(0).Q(1) \implies P(0).P(1)) \implies (Q(0) \implies P(0)).(Q(1) \implies P(1))$ Un contre modèle est $Q_I = \{0\}$ et $P_I = \{1\}$ ou $\emptyset$ (on a un contre modèle symétrique avec Q_I=$\{1\}$).
2. La 1-expansion est valide : $((0 = 0) \implies P(0,0)) \implies (P(0,0) \implies 0 = 0)$ pour toutes les interprétations de P.
   La 2-expansion est :
   $(((0 = 0) \implies P(0,0)) + ((0 = 1) \implies P(0,1)))$.
   $(((1 = 0) \implies P(1,0)) + ((1 = 1) \implies P(1,1))) \implies (P(0,0) \implies 0 = 0).(P(1,0) \implies 1 = 0)$.
   $(P(0,1) \implies 0 = 1).(P(1,1) \implies 1 = 1)$
   Un contre modèle est donnée par l'interprétation $P_I = \{(0,1),(1,0)\}$ (ou contenant un de ces deux couples).

Exercice 3, page 1

1. Utilisation de décomposition : $(1) x = f(y,a)$, $(2) u = z$, $(3) x = f(b,z)$.
   Elimination de (1) dans (3) donne (4) $f(y,a) = f(b,z)$.
   Utilisation de décomposition sur (4) donne (5) $y = b$, (6) $z = a$,
   on élimine z dans (2) en utilisant (6).
   On trouve $\sigma = \{x := f(b,a), y := b, u := a, z := a\}$.
2. Utilisation de décomposition : (1) $z = f(b,b)$, $(2) f(y, g(a,x)) = f(z, g(x, h(x,x)))$.
   Décomposition de (2) donne $(3) y = z$, (4) $g(a,x) = g(x, h(x,x))$.
   Décomposition de (4) donne $(5) x = a$, (6) $x = h(x,x)$.
   Échec d'élimination en (6) l'algorithme s'arrête pas d'unification possible.
3. Utilisation de décomposition : $(1) h(g(a,x), g(z,z)) = h(y,y)$, $(2) y = g(z,z)$.
   Décomposition de (1) donne $(3) g(a,x) = y$ et $(4) y = g(z,z)$.
   On peut supprimer (4) qui est la même équation que (2).
   On élimine y dans (3) avec (2) ce qui donne $(5) g(z,z) = g(a,x)$.
   Décomposition de (5) donne (6) $z = a$ et (7) $z = x$.
   On trouve $\sigma = \{z := a, x := a, y = g(a,a)\}$.

Exercice 4, page 2

1. — $\neg A \equiv \exists x \forall y Q(x,y) \wedge \forall x\big(\neg Q(x,x) \vee \exists y R(y,x)\big) \wedge \neg \exists y \exists x R(x,y)$
   — Forme normale : $\exists x \forall y Q(x,y) \wedge \forall x\big(\neg Q(x,x) \vee \exists y R(y,x)\big) \wedge \forall y \forall x \neg R(x,y)$
   — Forme propre : $\exists x \forall y Q(x,y) \wedge \forall z\big(\neg Q(z,z) \vee \exists t R(t,z)\big) \wedge \forall u \forall v \neg R(v,u)$
   — Élimination des $\exists$: $\forall y Q(a,y) \wedge \forall z\big(\neg Q(z,z) \vee R(f(z),z)\big) \wedge \forall u \forall v \neg R(v,u)$
   — Élimination des $\forall$: $\forall y \forall z \forall u \forall v\Big(Q(a,y) \wedge \big(\neg Q(z,z) \vee R(f(z),z)\big) \wedge \neg R(v,u)\Big)$
   — Forme clausale : $Q(a,y),\ \overline{Q(z,z)} + R(f(z),z),\ \overline{R(v,u)}$
2. On instancie $y := a, z := a, v := f(a), u := a$.
   Alors $Q(a,a)$ et $\overline{Q(a,a)} + R(f(a),a)$ ont pour résolvant $R(f(a),a)$ qui est contradictoire avec $\overline{R(f(a),a)}$.

Exercice 5, page 2

— On effectue $\{x := f(a), y := b\}$ dans $\overline{P(x)} + \overline{P(f(a))} + Q(y)$ ce qui donne $\overline{P(f(a))} + Q(b)$ on applique la résolution avec $P(y)$ instanciée avec $\{y := f(a)\}$ ce qui donne la résolvante (1) $Q(b)$.
On fait une autre instance de $P(y)$ avec $\{y := g(b,a)\}$ ce qui donne $P(g(b,a))$. En instantiant $\overline{P(g(b,x)), \overline{Q(b)}}$ avec $\{x := a\}$ on obtient $\overline{P(g(b,a)), \overline{Q(b)}}$. En appliquant la résolution à ces deux instantiations on obtient (2) $\overline{Q(b)}$.
On conclut en faisant une résolution sur (1) et (2) qui donne la clause vide.

— Par factorisation de $\overline{P(x)} + \overline{P(f(a))} + Q(y)$ on obtient $\overline{P(f(a))} + Q(y)$. Par résolution propositionnelle avec $P(y)$ on obtient $Q(y)$. Par résolution propositionnelle entre $P(y)$ et $\overline{P(g(b,x)) + \overline{Q(b)}}$ on obtient $Q(b)$ avec une dernière résolution propositionnelle entre $Q(y)$ et $\overline{Q(b)}$ on trouve la clause vide.

Exercice 6, page 2

1. *(10 points)* $\forall y \exists x \neg Q(x,y) \Rightarrow \neg \forall x \forall y Q(x,y)$

contexte	numéro	preuve	justification
1	1	`Supposons` $\forall y \exists x \neg Q(x,y)$	
1,2	2	`Supposons` $\forall x \forall y Q(x,y)$	
1,2	3	$\exists x \neg Q(x,y)$	$\forall E y$ 2
1,2,4	4	`Supposons` $\neg Q(x,y)$	
1,2,4	5	$\forall y Q(x,y)$	$\forall E x$ 4
1,2,4	6	$Q(x,y)$	$\forall E y$ 5
1,2,4	7	$\perp$	$\Rightarrow E$ 4,6
1,2	8	`Donc` $\neg Q(x,y) \Rightarrow \perp$	$\Rightarrow I$ 4, 7
1,2	9	$\perp$	$\exists E$ 3,8
	10	`Donc` $\neg \forall x \forall y Q(x,y)$	$\Rightarrow I$ 2, 9
	11	`Donc` $\forall y \exists x \neg Q(x,y) \Rightarrow \neg \forall x \forall y Q(x,y)$	$\Rightarrow I$ 1, 10

2. *(10 points)* $\forall x \forall y (P(x) \wedge \neg P(y) \Rightarrow x \neq y)$

contexte	numéro	preuve	justification
1	1	`Supposons` $P(x) \wedge \neg P(y)$	
1,2	2	`Supposons` $x = y$	
1,2	3	$P(x)$	$\wedge E1$ 1
1,2	4	$\neg P(y)$	$\wedge E2$ 1
1,2	5	$P(y)$	*cong* 2,3
1,2	6	$\perp$	$\Rightarrow E$ 4,5
1	7	`Donc` $x \neq y$	$\Rightarrow I$ 2,6
	8	`Donc` $P(x) \wedge \neg P(y) \Rightarrow x \neq y$	$\Rightarrow I$ 1,7
	9	$\forall y (P(x) \wedge \neg P(y) \Rightarrow x \neq y)$	$\forall I$ 8
	10	$\forall x \forall y (P(x) \wedge \neg P(y) \Rightarrow x \neq y)$	$\forall I$ 9

Exercice 7, page 2

1. Montrons que $|$, définie par $x \mid y = 1$ si et seulement si $x = 0$ et $y = 0$, est complète.
   Les connecteurs $\{\neg, \bot, \vee, \wedge\}$ sont définissables à l'aide de $|$:
   — $\neg x = x \mid x$
   — $\bot = \neg x \mid x = (x \mid x) \mid x$
   — $x \vee y = \overline{x \mid y} = (x \mid y) \mid (x \mid y)$
   — $x \wedge y = \bar{x} \mid \bar{y} = (x \mid x) \mid (y \mid y)$
   — enfin on exprime $\top$ comme $\neg\bot = ((x \mid x) \mid x) \mid ((x \mid x) \mid x)$
2. Montrons que l'ensemble $\{1, \Rightarrow\}$ est incomplet.
   Appelons stable pour 1, une fonction booléenne qui prend la valeur 1 quand tous ses arguments ont la valeur 1.
   Il est facile de voir que toute fonction booléenne définie avec les seules opérations $\{1, \Rightarrow\}$ est stable pour 1, car $1 \Rightarrow 1 \equiv 1$ (on peut le faire proprement par récurrence sur la taille de la formule).
   Or $\neg$ n'est pas stable pour 1, donc la fonction $f(x) = \neg x$ n'est pas exprimable.
   Par suite cet ensemble est incomplet.

Partiel INF402

Eskandar Kouicem Benjamin Wack

14 mars 2022
2 pages
Total : 120 points + 5 points de bonus
Durée : 2h00

Documents autorisés : une feuille recto verso de notes manuscrites format A4.
Le barème est *indicatif*, les points correspondent au nombre de minutes nécessaires pour réaliser les exercices. L'épreuve sera notée sur 120 points + 5 points de bonus.
Le résultat d'une question peut être admis pour s'en servir dans la suite de l'énoncé. Les exercices peuvent être traités dans l'ordre de votre choix à condition de les numéroter clairement.

Exercice 1 (Modèles et formes normales (25 points))
Soit $A = a \vee \neg c \Leftrightarrow d \Rightarrow b \wedge d \vee \neg a \Rightarrow b$.

1. *Réécrivez cette formule sous forme d'arbre.* *(3 points)*
2. *Déterminer une forme normale disjonctive de A et la simplifier le plus possible.* *(9 points)*
3. *Déterminer une forme normale conjonctive de A et la simplifier le plus possible.* *(9 points)*
4. *Si c'est possible, donner un modèle de A et justifier la réponse* sans utiliser de table de vérité. *(2 points)*
5. *Si c'est possible, donner un contre-modèle de A et justifier la réponse* sans utiliser de table de vérité. *(2 points)*

Réponse:
- Je donne A sous sa forme stricte : $(a \vee \neg c) \Leftrightarrow (d \Rightarrow (((b \wedge d) \vee \neg a) \Rightarrow b))$.
- En forme normale disjonctive : $a + c.d.\bar{b} + \bar{c}.\bar{d} + \bar{c}.b$
  Ce qui donne 4 modèles : (1) $a = 0$, (2) $b = 0, c = d = 1$, (3) $c = d = 0$, (4) $b = 1, c = 0$.
- En forme normale conjonctive : $(a + \bar{c} + d).(a + \bar{c} + \bar{b}).(a + c + \bar{d} + b)$
  Ce qui donne 3 contre-modèles : (1) $a = d = 0, c = 1$, (2) $a = 0, b = c = 0$, (3) $a = b = c = 0, d = 1$.

□

Exercice 2 (Formalisation (Exo du poly, 15 points))
Formaliser les énoncés suivants en logique propositionnelle :

(a) *Si Pierre est rentré chez lui, alors Jean est allé au cinéma.*

(b) *Marie est à la bibliothèque ou Pierre est rentré chez lui.*

(c) *Si Jean est allé au cinéma, alors Marie est à la bibliothèque ou Pierre est rentré chez lui.*

(d) *Marie n'est pas à la bibliothèque et Jean est allé au cinéma.*

(e) *Pierre est rentré chez lui.*

À l'aide d'une des méthodes vues en cours (au choix), déterminer si le dernier énoncé (e) est conséquence des énoncés (a) (b) (c) et (d).
Réponse: Les faits, Pierre est rentré chez lui, Jean est allé au cinéma, Marie est à la bibliothèque sont dénotés respectivement par les variables p, j, m. Les hypothèses sont formalisées par les formules :

1. $p \Rightarrow j$.
2. $m \vee p$.
3. $j \Rightarrow m \vee p$.

4. $\neg m \wedge j$.

Et la conclusion par p.

En utilisant les équivalences remarquables, nous prouvons que $(p \Rightarrow j) \wedge (m \vee p) \wedge (j \Rightarrow m \vee p) \wedge (\neg m \wedge j) \wedge \neg p$ est contradictoire. Car en enlevant les implications nous avons :

$$(\neg p \vee j) \wedge (m \vee p) \wedge (\neg j \vee m \vee p) \wedge \neg m \wedge j \wedge \neg p.$$

En utilisant $(x \vee \neg y) \wedge y = x \wedge y$, entre la seconde et la quatrième clause nous obtenons :

$$(\neg p \vee j) \wedge p \wedge (\neg j \vee m \vee p) \wedge \neg m \wedge j \wedge \neg p.$$

Nous obtenons une contradiction entre p et $\neg p$. Et donc le raisonnement est correct.

□

Exercice 3 (Résolution (20 points))

Montrez **par résolution** *que les ensembles de formules suivants sont insatisfaisables :*

1. $\{a + \bar{c}, c, \bar{a} + b + \bar{d} + e, b + d, e + \bar{b}, f + g, \bar{g} + f, \bar{f} + \bar{e}\}$ *(10 points)*
2. $\{\bar{q} + m, \bar{q} + \bar{m} + \bar{r}, \bar{q} + \bar{m} + r, \bar{p} + r, \bar{p} + \bar{r}, p + q\}$ *(10 points)*

Réponse:

1	$a + \bar{c}$	Hyp
2	c	Hyp
3	a	Résolvant de 1, 2
4	$\bar{a} + b + \bar{d} + e$	Hyp
5	$b + \bar{d} + e$	Résolvant de 3, 4
6	$b + d$	Hyp
7	$b + e$	Résolvant de 5, 6
8	$e + \bar{b}$	Hyp
9	e	Résolvant 7,8
10	$f + g$	Hyp
11	$\bar{g} + f$	Hyp
12	f	Résolvant de 10,11
13	$\bar{f} + \bar{e}$	Hyp
14	$\bar{e}$	Résolvant de 12,13
15	$\perp$	Résolvant de 9,14

1	$\bar{q} + \bar{m} + \bar{r}$	Hyp
2	$\bar{q} + \bar{m} + r$	Hyp
3	$\bar{q} + \bar{m}$	Résolvant de 1, 2
4	$\bar{q} + m$	Hyp
5	$\bar{q}$	Résolvant de 3, 4
6	$p + q$	Hyp
7	p	Résolvant de 5, 6
8	$\bar{p} + r$	Hyp
9	$\bar{p} + \bar{r}$	Hyp
10	$\bar{p}$	Résolvant de 8,9
11	$\perp$	Résolvant de 7,10

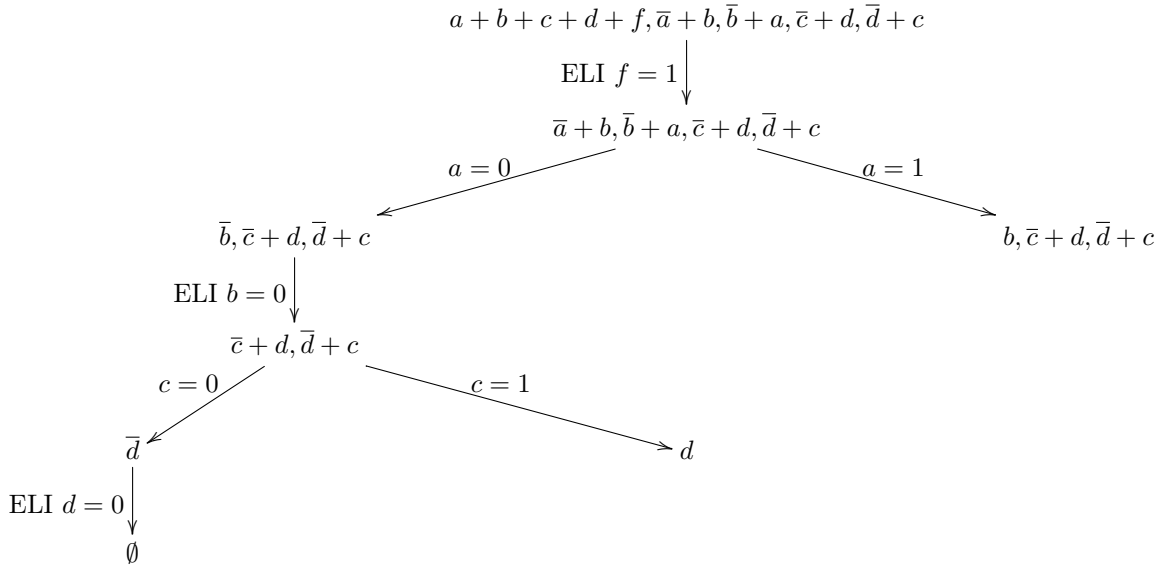
□

Exercice 4 (DPLL (20 points))
Déterminez **en appliquant l'algorithme DPLL** *si les ensembles de clauses suivants sont satisfaisables. Le cas échéant, vous donnerez le modèle exhibé par DPLL.*
Vous donnerez une trace complète du déroulement de l'algorithme étiquetée par les règles utilisées.

1. $\{a+b+c+d+f, \overline{a}+b, \overline{b}+a, \overline{c}+d, \overline{d}+c\}$. *(10 points)*
2. $\{p+q+\bar{r},\ r+q+\bar{p},\ p+r+q,\ r+q,\ q+p,\ \bar{p}+\bar{q},\ \bar{q}+p,\ q+s,\ \bar{s}+\bar{p}\}$ *(10 points)*

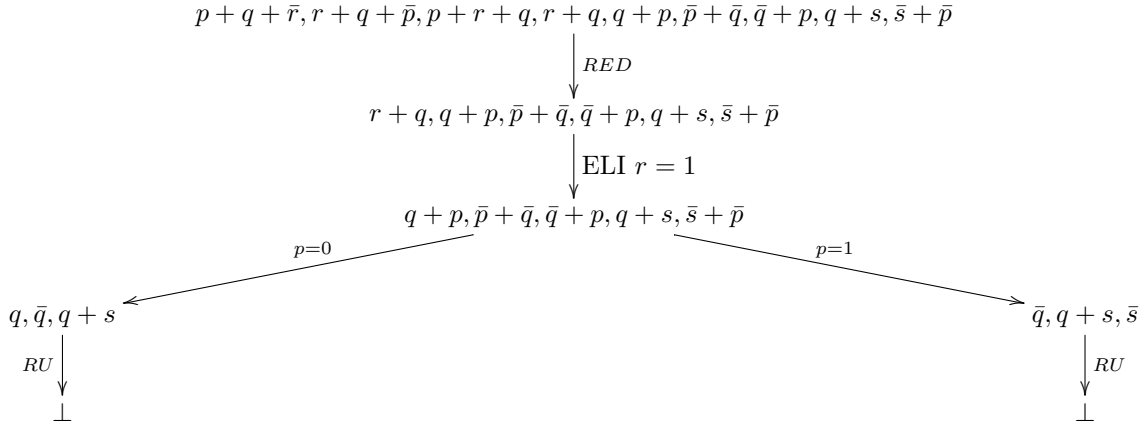
Réponse:

1.



D'après la trace ci-dessus, l'ensemble est satisfaisable. Nous observerons que, d'après le sens de « ou alors », l'algorithme se termine sur la première feuille, où la valeur vrai est retourné. Un modèle est donc $d=0$, $c=1$, $b=0$, $a=0$, et $f=1$.

2.



□

Exercice 5 (Déduction naturelle (25 points))
Prouver, **par déduction naturelle**, *les formules suivantes.*

Chaque formule de la preuve devra être justifiée ; il vous est recommandé de présenter vos preuves sous forme de tableaux comme vu en cours.

1. $((p \vee q) \Rightarrow r) \Rightarrow (p \Rightarrow r)$ *(5 points)*
2. $(p \Rightarrow \neg p) \wedge (\neg p \Rightarrow p) \Rightarrow \perp$ *(10 points)*
3. $p \Rightarrow (\neg p \vee q) \Rightarrow q$ *(10 points)*

Réponse:

1. $((p \vee q) \Rightarrow r) \Rightarrow (p \Rightarrow r)$

```
1    assume (p + q) => r.
2      assume p.
3        p + q.                                          +I1 2
4        r.                                              =>E 1,3
5      therefore p => r.                                 =>I 2,4
6    therefore ((p + q) => r) => (p => r).               =>I 1,5
```

2. $(p \Rightarrow \neg p) \wedge (\neg p \Rightarrow p) \Rightarrow \bot$

```
1    assume (p => -p) & (-p => p).
2      p => -p.                                          &E1 1
3      -p => p.                                          &E2 1
4      assume p.
5        -p.                                             =>E 2,4
6        F.                                              =>E 5,4
7      therefore -p.                                     =>I 4,6
8      p.                                                =>E 3,7
9      F.                                                =>E 7,8
10   therefore (p => -p) & (-p => p) => F.               =>I 1,9
```

3. $p \Rightarrow (\neg p \vee q) \Rightarrow q$

```
1    assume p.
2      assume -p + q.
3        assume -p.
4          F.                                            =>E 3,1
5          q.                                            Efq 4
6        therefore -p => q.                              =>I 3,5
7        assume q.
8        therefore q => q.                               =>I 7,7
9        q.                                              +E 2,6,8
10     therefore -p + q => q.                            =>I 2,9
11   therefore p => -p + q => q.                         =>I 1,10
```

□

Exercice 6 (Démonstration par récurrence (20 points))
Démontrer que toute formule stricte construite uniquement avec :

- *la constante* $\top$,
- *la variable propositionnelle* x,
- *les connecteurs* $\Rightarrow$ *et* $\vee$,
- *et les parenthèses*

est équivalente soit à x *soit à* $\top$.

Réponse: Par récurrence sur la taille de la formule.

Une formule de taille 0 est nécessairement une variable ou une constante, donc en l'occurrence x ou $\top$.

Soit un entier $n > 0$, supposons la propriété vraie pour toute formule de taille strictement inférieure à n. Soit alors A une formule de taille n de la forme décrite dans l'énoncé. A est forcément de la forme $B \Rightarrow C$ ou bien $B \vee C$, où B et C sont des formules de taille strictement inférieure à n, toutes deux construites comme A.

Par hypothèse de récurrence, B et C sont toutes deux équivalentes à x ou $\top$, on est donc ramené à étudier 8 cas :

- $A \equiv x \Rightarrow x \equiv \top$
- $A \equiv x \Rightarrow \top \equiv \top$
- $A \equiv \top \Rightarrow x \equiv x$
- $A \equiv \top \Rightarrow \top \equiv \top$
- $A \equiv x \vee x \equiv x$
- $A \equiv \top \vee x \equiv \top$ et de même pour $x \vee \top$ et $\top \vee \top$

□

Examen INF402

Eskandar Kouicem Benjamin Wack

23 mai 2022
2 pages
Total : 120 points
Durée : 2h00

Documents autorisés : une feuille recto verso de notes manuscrites format A4.
Le barème est *indicatif*, les points correspondent au nombre de minutes nécessaires pour réaliser les exercices. L'épreuve sera notée sur 120 points.
Le résultat d'une question peut être admis pour s'en servir dans la suite de l'énoncé. Les exercices peuvent être traités dans l'ordre de votre choix à condition de les numéroter clairement.

Exercice 1 (Expansions (20 points))**.**
À l'aide de la méthode des expansions, déterminer ***un contre-modèle*** *de chacune de ces formules :*

1. (10 points) $A = \exists x\big(P(x) \Rightarrow \forall y(P(y) \Rightarrow x = y)\big)$.
2. (10 points) $B = \forall x \exists y R(x, y) \Rightarrow \exists y \forall x R(x, y)$.

□

Exercice 2 (Formalisation (20 points, exercice du poly))**.**
Nous considérons $\Sigma = \{f^{r2}, o^{r2}, c^{r2}, j^{r2}, r^{f0}, p^{f1}\}$ *la signature ayant la sémantique donnée ci-dessous.*
— $f(x, y)$: *x est un frère de y.*
— $o(x, y)$: *x est un oncle de y.*
— $c(x, y)$: *x est un cousin de y.*
— $j(x, y)$: *x est plus jeune que y.*
— *r est le diminutif de Robert.*
— $p(x)$ *est le père de x.*

1. (12 points) *Exprimer en logique du premier ordre et en utilisant la signature Σ les phrases suivantes :*
   (a) *Tout frère du père de Robert est un oncle de Robert.*
   (b) *Si les pères de deux enfants sont des frères alors ces deux enfants sont des cousins.*
   (c) *Robert a un cousin plus jeune qu'un des frères de Robert.*
2. (8 points) *Exprimer en français les propositions logiques suivantes :*
   (a) $\neg\exists x\ j(p(x), x)$
   (b) $\forall x \forall y(p(p(x)) = p(p(y)) \Rightarrow c(x, y))$
   (c) $\forall x \exists y(f(x, y) \wedge j(x, y))$
   (d) $\exists x \exists y(f(x, y) \wedge \neg(p(x) = p(y)))$

□

Exercice 3 (Unification (15 points))**.**
Pour chacune des équations suivantes, déterminez si elle admet une solution, et si oui déterminez un unificateur le plus général. Vous utiliserez pour cela l'algorithme du cours dont vous détaillerez les étapes en donnant à chaque fois le nom des règles utilisées.
a et b sont des constantes ; f, g et h sont des fonctions.

1. (5 points) $f(x, g(a), x) = f(y, g(y), b)$
2. (5 points) $f(f(x, y, a), g(y), h(y, b)) = f(z, x, h(b, b))$
3. (5 points) $h(g(x), h(y, x)) = h(g(y), h(g(z), z))$

□

Exercice 4 (Skolémisation (15 points)).

Soit A la formule suivante : $\neg\forall x P(x) \vee \neg\forall x Q(x) \Rightarrow \neg(\forall x P(x) \wedge \forall x Q(x))$

Montrez que A est valide en montrant que $\neg A$ est contradictoire par la méthode de Herbrand, en suivant les étapes suivantes :

1. (10 points) *Skolémisez $\neg A$ puis mettez-la en forme clausale.*
2. (5 points) *Proposez des instances contradictoires des clauses obtenues et montrez la contradiction par résolution propositionnelle.*

□

Exercice 5 (Résolution (15 points)).

Considérons l'ensemble suivant de clauses du premier ordre :

$$\Gamma \;=\; \{\overline{C(x,y)} + \overline{E(z,x)} + E(z,y),\ C(c,a),\ C(a,b),\ \overline{C(c,b)},\ C(x,y) + E(x,x),\ C(x,y) + \overline{E(x,y)}\}$$

Nous voulons montrer que cet ensemble de clauses est insatisfaisable par instanciation et par résolution.

1. (7 points) *Trouvez des instances contradictoires de ces clauses et montrez par résolution propositionnelle que ces instances sont contradictoires.*
2. (8 points) *Donnez une preuve directe du caractère contradictoire de Γ par factorisation, copie et résolution binaire. Il est possible que toutes les règles ne soient pas utilisées.*

□

Exercice 6 (Déduction naturelle (20 points)).

Démontrez les formules suivantes en déduction naturelle au premier ordre. Comme vu en cours et en TD, on numérotera les lignes de la preuve, et on indiquera pour chaque ligne la formule démontrée, le contexte, la règle appliquée et les prémisses utilisées.

1. (10 points) $\neg\exists x(P(x) \vee Q(x)) \Rightarrow \forall x(\neg P(x) \wedge \neg Q(x))$
2. (10 points) $\forall x \forall y(\exists z(x = z \wedge z = y) \Rightarrow (x = y))$

□

Exercice 7 (Démonstration par récurrence (15 points)).

Soit n un entier positif non nul. Soient $A_1, A_2, \ldots A_n$ et B des formules du premier ordre.

On suppose que x_1 est la seule variable libre dans A_1, que x_2 est la seule variable libre dans A_2, etc. et que y est la seule variable libre dans B.

On s'intéresse à la formule $\exists x_n A_n \Rightarrow \ldots \Rightarrow \exists x_2 A_2 \Rightarrow \exists x_1 A_1 \Rightarrow \forall y B$.

1. (1 point) *Rappeler quelle est la formule stricte (parenthésée) correspondant à cette formule.*
2. (14 points) *Démontrer que, pour tout $n \geqslant 1$, on a*

$$\exists x_n A_n \Rightarrow \ldots \Rightarrow \exists x_2 A_2 \Rightarrow \exists x_1 A_1 \Rightarrow \forall y B \quad \equiv \quad \forall x_n \ldots \forall x_2 \forall x_1 \forall y(A_n \Rightarrow \ldots \Rightarrow A_2 \Rightarrow A_1 \Rightarrow B)$$

□

Corrigés

Exercice 1, page 1

1. La 1-expansion est valide : $P(0) \Rightarrow (P(0) \Rightarrow 0 = 0)$
   La 2-expansion est :

$$\begin{aligned} & \big(P(0) \Rightarrow (P(0) \Rightarrow 0 = 0).(P(1) \Rightarrow 0 = 1)\big) + \big(P(1) \Rightarrow (P(0) \Rightarrow 1 = 0).(P(1) \Rightarrow 1 = 1)\big) \\ \equiv \quad & \big(P(0) \Rightarrow (P(1) \Rightarrow \bot)\big) + \big(P(1) \Rightarrow (P(0) \Rightarrow \bot)\big) \\ \equiv \quad & (P(0) \Rightarrow \overline{P(1)}) + (P(1) \Rightarrow \overline{P(0)}) \;\equiv\; P(0) \Rightarrow \overline{P(1)} \end{aligned}$$

   Le (seul) contre-modèle est donc $P_I = \{0, 1\}$.
2. La 1-expansion est valide : $R(0,0) \Rightarrow R(0,0)$
   La 2-expansion est :
   $(R(0,0) + R(0,1)).(R(1,0) + R(1,1)) \Rightarrow R(0,0).R(1,0) + R(0,1).R(1,1)$.
   Les contre-modèles sont donc $R_I = \{(0,0),\ (1,1)\}$ et $R_I = \{(0,1),\ (1,0)\}$.

Exercice 2, page 1

1. Tout frère du père de Robert est un oncle de Robert :

$$\forall x(f(x, p(r)) \Rightarrow o(x, r)).$$

2. Si les pères de deux enfants sont des frères alors ces deux enfants sont des cousins :

$$\forall x \forall y(f(p(x), p(y)) \Rightarrow c(x, y)).$$

3. Robert a un cousin plus jeune qu'un des frères de Robert :

$$\exists x(c(x, r) \wedge \exists y(f(y, r) \wedge j(x, y))).$$

1. $\neg\exists x\ j(p(x), x)$: personne n'est plus âgé que son père
2. $\forall x \forall y(p(p(x)) = p(p(y)) \Rightarrow c(x, y))$: deux personnes qui ont le même grand-père paternel sont cousines
3. $\forall x \exists y(f(x, y) \wedge j(x, y))$: tout le monde a un frère aîné (impossible mais ça n'empêche pas de l'exprimer...)
4. $\exists x \exists y(f(x, y) \wedge \neg(p(x) = p(y)))$: il y a deux frères qui n'ont pas le même père

Exercice 3, page 1

1. $f(x, g(a), x) = f(y, g(y), b)$
   Décomposition : $x = y$, $g(a) = g(y)$, $x = b$
   Élimination de x : $x := y$, $g(a) = g(y)$, $y = b$
   Décomposition : $x := y$, $a = y$, $y = b$
   Élimination de y : $x := y$, $a = b$, $y := b$
   Échec de décomposition : pas de solution.
2. $f(f(x, y, a), g(y), h(y, b)) = f(z, x, h(b, b))$
   Décomposition : $f(x, y, a) = z$, $g(y) = x$, $h(y, b) = h(b, b)$
   Orientation ($\times 2$) : $z = f(x, y, a)$, $x = g(y)$, $h(y, b) = h(b, b)$
   Élimination de z : $z := f(x, y, a)$, $x = g(y)$, $h(y, b) = h(b, b)$
   Élimination de x : $z := f(g(y), y, a)$, $x := g(y)$, $h(y, b) = h(b, b)$
   Décomposition : $z := f(g(y), y, a)$, $x := g(y)$, $y = b$, $b = b$

Identité : $z := f(g(y), y, a)$, $x := g(y)$, $y = b$

Élimination de y : $z := f(g(b), b, a)$, $x := g(b)$, $y := b$

$< x := g(b), y := b, z := f(g(b), b, a) >$ est une solution la plus générale de l'équation.

3. $h(g(x), h(y, x)) = h(g(y), h(g(z), z))$

   Décomposition : $g(x) = g(y)$, $h(y, x) = h(g(z), z))$

   Décomposition : $x = y$, $h(y, x) = h(g(z), z))$

   Élimination de x : $x := y$, $h(y, y) = h(g(z), z))$

   Décomposition : $x := y$, $y = g(z)$, $y = z$

   Élimination de y : $x := y$, $y := g(z)$, $g(z) = z$

   Orientation : $x := y$, $y := g(z)$, $z = g(z)$

   Échec d'élimination de z : pas de solution.

Exercice 4, page 2

$\neg A = \neg(\neg\forall x P(x) \vee \neg\forall x Q(x) \Rightarrow \neg(\forall x P(x) \wedge \forall x Q(x)))$

1. Forme normale : $(\exists x \neg P(x) \vee \exists x \neg Q(x)) \wedge \forall x P(x) \wedge \forall x Q(x))$
2. Forme propre : $(\exists x \neg P(x) \vee \exists y \neg Q(y)) \wedge \forall u P(u) \wedge \forall v Q(v))$
3. Elimination des $\exists$: $(\neg P(a) \vee \neg Q(b)) \wedge \forall u P(u) \wedge \forall v Q(v))$
4. Enlèvement des $\forall$: $(\neg P(a) \vee \neg Q(b)) \wedge P(u) \wedge Q(v))$
5. Forme clausale : $\{\neg P(a) \vee \neg Q(b), P(u), Q(v)\}$
6. Instances contradictoires : $\neg P(a) \vee \neg Q(b), P(a), Q(b)$

1 :	$\neg P(a) \vee \neg Q(b)$	Hyp
2 :	$P(a)$	Hyp
3 :	$\neg Q(b)$	Res 1,2
4 :	$Q(b)$	Hyp
5 :	$\perp$	Res 3,4

Exercice 5, page 2

Instanciation : Nous obtenons les instances contradictoires suivantes :

$$\begin{aligned}\Gamma \quad = \quad & \{\overline{C(c,a)} + \overline{E(c,c)} + E(c,a), \overline{C(a,b)} + \overline{E(c,a)} + E(c,b), \quad \text{(deux instances)} \\ & C(c,a), C(a,b), \overline{C(c,b)}, E(c,c) + C(c,b), \overline{E(c,b)} + C(c,b)\}\end{aligned}$$

Par résolution propositionnelle nous avons la preuve :

(1)	$\overline{C(c,b)}$	Hyp
(2)	$E(c,c) + C(c,b)$	Hyp
(3)	$\overline{E(c,b)} + C(c,b)$	Hyp
(4)	$E(c,c)$	Res 1,2
(5)	$\overline{E(c,b)}$	Res 1,3
(6)	$\overline{C(c,a)} + \overline{E(c,c)} + E(c,a)$	Hyp
(7)	$\overline{C(a,b)} + \overline{E(c,a)} + E(c,b)$	Hyp
(8)	$\overline{C(c,a)} + E(c,a)$	Res 4,6
(9)	$\overline{C(a,b)} + \overline{E(c,a)}$	Res 5,7
(10)	$\overline{C(c,a)} + \overline{C(a,b)}$	Res 8,9
(11)	$C(c,a)$	Hyp
(12)	$\overline{C(a,b)}$	Res 10,11
(13)	$C(a,b)$	Hyp
(14)	$\perp$	Résolvant 12,13

Résolution binaire :

numéro	conclusion	justification	unificateur
(1)	$\overline{C(c,b)}$	Hyp	
(2)	$\overline{E(x,x)} + C(x,y)$	Hyp	
(3)	$\overline{E(x,y)} + C(x,y)$	Hyp	
(4)	$\overline{E(c,c)}$	Res 1,2	$< x := c, y := b >$
(5)	$\overline{E(c,b)}$	Res 1,3	$< x := c, y := b >$
(6)	$\overline{C(x,y)} + \overline{E(z,x)} + E(z,y)$	Hyp	
(7)	$\overline{C(c,y)} + \overline{E(c,y)}$	Res 4,6	$< x := c, z := c >$
(8)	$\overline{C(x,b)} + \overline{E(c,x)}$	Res 5,6	$< y := b, z := c >$
(9)	$\overline{C(c,y)} + \overline{C(y,b)}$	Res 7,8	$< x := y >$
(10)	$\overline{C(c,a)}$	Hyp	
(11)	$\overline{C(a,b)}$	Res 9,10	$< y := a >$
(12)	$C(a,b)$	Hyp	
(13)	$\perp$	Résolvant 11,12	

Exercice 6, page 2

1. $\neg\exists x(P(x) \vee Q(x)) \Rightarrow \forall x(\neg P(x) \wedge \neg Q(x))$

contexte	numéro	preuve	justification
1	1	`Supposons` $\neg\exists x(P(x) \vee Q(x))$	
1,2	2	`Supposons` $P(x)$	
1,2	3	$P(x) \vee Q(x)$	$\vee I1$ 2
1,2	4	$\exists x(P(x) \vee Q(x))$	$\exists I$ 3 x
1,2	5	$\perp$	$\Rightarrow E$ 1, 4
1	6	`Donc` $\neg P(x)$	$\Rightarrow I$ 2, 5
1,7	7	`Supposons` $Q(x)$	
1,7	8	$P(x) \vee Q(x)$	$\vee I2$ 7
1,7	9	$\exists x(P(x) \vee Q(x))$	$\exists I$ 8 x
1,7	10	$\perp$	$\Rightarrow E$ 1, 9
1	11	`Donc` $\neg Q(x)$	$\Rightarrow I$ 7, 10
1	12	$\neg P(x) \wedge \neg Q(x)$	$\wedge I$ 6, 11
1	13	$\forall x(\neg P(x) \wedge \neg Q(x))$	$\forall I$ 12
	14	`Donc` $\neg\exists x(P(x) \vee Q(x)) \Rightarrow \forall x(\neg P(x) \wedge \neg Q(x))$	$\Rightarrow I$ 1, 13

2. $\forall x \forall y(\exists z(x = z \wedge z = y) \Rightarrow (x = y))$

contexte	numéro	preuve	règle
1	1	`Supposons` $\exists z(x = z \wedge z = y)$	
1,2	2	`Supposons` $(x = z \wedge z = y)$	
1,2	3	$x = z$	$\wedge$E1 2
1,2	4	$z = y$	$\wedge$E2 2
1,2	5	$x = y$	congruence 4,3
1	6	`Donc` $(x = z \wedge z = y) \Rightarrow (x = y)$	$\Rightarrow$I 2,5
1	7	$x = y$	$\exists$E 1,6
	8	`Donc` $\exists z(x = z \wedge z = y) \Rightarrow (x = y)$	$\Rightarrow$I 1,7
	9	$\forall y(\exists z(x = z \wedge z = y) \Rightarrow (x = y))$	$\forall$I 8
	10	$\forall x \forall y(\exists z(x = z \wedge z = y) \Rightarrow (x = y))$	$\forall$I 9

3.

Exercice 7, page 2

1. $\Big(\exists x_n A_n \Rightarrow \Big(\ldots \Rightarrow \big(\exists x_2 A_2 \Rightarrow (\exists x_1 A_1 \Rightarrow \forall y B)\big)\Big)\Big)$.
2. Par récurrence (directe) sur n :

— Pour $n = 0$ on a trivialement $\forall y B \equiv \forall y B$
— Supposons $n \geq 1$ et la propriété vraie pour $n - 1$. Alors :

$$\begin{aligned}
\exists x_n A_n \Rightarrow \ldots \Rightarrow \exists x_2 A_2 \Rightarrow \exists x_1 A_1 \Rightarrow \forall y B \quad &\equiv \quad \neg(\exists x_n A_n) \vee (\exists x_{n-1} A_{n-1} \Rightarrow \ldots \Rightarrow \exists x_1 A_1 \Rightarrow \forall y B) \\
&\equiv \quad (\forall x_n \neg A_n) \vee (\exists x_{n-1} A_{n-1} \Rightarrow \ldots \Rightarrow \exists x_1 A_1 \Rightarrow \forall y B)
\end{aligned}$$

Or la formule $\exists x_{n-1} A_{n-1} \Rightarrow \ldots \Rightarrow \exists x_1 A_1 \Rightarrow \forall y B$ satisfait la propriété au rang $n - 1$:

$$\equiv \quad (\forall x_n \neg A_n) \vee \forall x_{n-1} \ldots \forall x_1 \forall y (A_{n-1} \Rightarrow \ldots \Rightarrow A_1 \Rightarrow B)$$

Comme les variables ne sont pas libres dans les autres parties de la formule :

$$\begin{aligned}
&\equiv \quad \forall x_n \forall x_{n-1} \ldots \forall x_1 \forall y \big(\neg A_n \vee (A_{n-1} \Rightarrow \ldots \Rightarrow A_1 \Rightarrow B)\big) \\
&\equiv \quad \forall x_n \forall x_{n-1} \ldots \forall x_1 \forall y (A_n \Rightarrow A_{n-1} \Rightarrow \ldots \Rightarrow A_1 \Rightarrow B)
\end{aligned}$$

Partiel INF452

Stéphane Devismes Benjamin Wack

10 mars 2021
2 pages
Total : 120 points + 5 points de bonus
Durée : 2h00

Documents autorisés : une feuille recto verso de notes manuscrites format A4.
Le barème est *indicatif*, les points correspondent au nombre de minutes nécessaires pour réaliser les exercices.
L'épreuve sera notée sur 120 points + 5 points de bonus.
Le résultat d'une question peut être admis pour s'en servir dans la suite de l'énoncé. Les exercices peuvent être traités dans l'ordre de votre choix à condition de les numéroter clairement.

Exercice 1 (Modèles et formes normales (20 points))
Soit $A = a \wedge d \Rightarrow b \wedge c \Leftrightarrow \neg c \Rightarrow \neg d \wedge a$.

1. *Réécrivez cette formule sous forme d'arbre.* *(1 point)*
2. *Déterminer une forme normale disjonctive de A et la simplifier le plus possible.* *(8 points)*
3. *Déterminer une forme normale conjonctive de A et la simplifier le plus possible.* *(8 points)*
4. *Si c'est possible, donner un modèle de A et justifier la réponse* sans utiliser de table de vérité. *(1,5 points)*
5. *Si c'est possible, donner un contre-modèle de A et justifier la réponse* sans utiliser de table de vérité. *(1,5 points)*

Réponse:
— Je donne A sous sa forme stricte : $(((a \wedge d) \Rightarrow (b \wedge c)) \Leftrightarrow (\neg c \Rightarrow (\neg d \wedge a)))$.
— En forme normale disjonctive : $a \wedge \neg d \vee a \wedge b \vee a \wedge \neg c \vee \neg a \wedge c$
Ce qui donne quatre modèles : (1) $a = 1, d = 0$, (2) $a = 1, b = 1$, (3) $a = 1, c = 0$ et (4) $a = 0$ et $c = 1$.
— En forme normale conjonctive : $(\neg a \vee b \vee \neg c \vee \neg d) \wedge (a \vee c)$
Ce qui donne deux contre-modèles : (1) $a = 1, b = 0, c = 1, d = 1$ et (2) $a = 0, c = 0$.

□

Exercice 2 (Formalisation (25 points))

Nous retournons sur l'île des Purs et des Pires[1]*. Rappelons que les Purs disent toujours la vérité tandis que les Pires mentent toujours.*

Nous rencontrons cette fois trois indigènes, Aha, Bébé et Sissi. Nous savons que l'un d'entre eux est Pur alors que les deux autres sont Pires. Ils affirment :

Aha : *« Il n'y a aucun danger sur cette île. »*

Bébé : *« Aha est Pur. »*

Sissi : *« Bébé est Pire. »*

Peut-on en conclure que l'île est dangereuse ?

1. *Formaliser les trois affirmations, la conclusion, et le fait qu'il y a un Pur et deux Pires. (10 points)*
Pour cela, vous utiliserez les variables propositionnelles suivantes :
— *A : Aha est Pur.*
— *B : Bébé est Pur.*
— *C : Sissi est Pur.*

1. Cette énigme est librement inspirée d'un problème de Jean-Claude Baillif et réutilise les personnages de Raymond Smullyan.

— *D : L'île est dangereuse.*

2. *Transformer chacune des trois affirmations, le fait qu'il y a un Pur et deux Pires, ainsi que la négation de la conclusion en un ensemble de clauses équivalentes. (10 points)*
3. *Démontrer,* **par résolution**, *que la conclusion se déduit des trois affirmations. (5 points)*

Réponse:

1. **Affirmation de Aha :** $A \Leftrightarrow \neg D$

   Affirmation de Bébé : $B \Leftrightarrow A$

   Affirmation de Sissi : $C \Leftrightarrow \neg B$

   Conclusion : D

   Tribus des personnages : par exemple $A \wedge \neg B \wedge \neg C + \neg A \wedge \neg B \wedge C + \neg A \wedge B \wedge \neg C$
2. Clauses :
   — $\bar{A} + \bar{D}$, $A + D$
   — $\bar{B} + A$, $\bar{A} + B$
   — $\bar{B} + \bar{C}$, $B + C$
   — $\bar{D}$
   — $A + B + C$, $\bar{A} + \bar{B}$, $\bar{B} + \bar{C}$, $\bar{A} + \bar{C}$
3. On n'utilise en fait que très peu de clauses, et Sissi ne nous apprend rien :

```
1 -D      hyp
2 A+D     hyp
3 A       res 1,2
4 -A+B    hyp
5 -A+-B   hyp
6 -A      res 4,5
7 Faux    res 3,6
```

□

Exercice 3 (Résolution (Exo du Poly, 20 points))

Les ensembles de formules suivants sont insatisfaisables.

— $\{a + b, \overline{a} + c, \overline{a} + \overline{d}, d + \overline{c}, \overline{b} + a\}$.

— $\{a + b + c, \overline{a} + b, \overline{b} + c, \overline{c} + a, \overline{a} + \overline{b} + \overline{c}\}$.

En donner une preuve par résolution.

Réponse:

— Soit l'ensemble de clauses : $\{a + b, \overline{a} + c, \overline{a} + \overline{d}, d + \overline{c}, \overline{b} + a\}$. Nous suggérons une tactique pour obtenir une preuve : construire en priorité les résolvants ayant le moins de littéraux possibles.

1	$a + b$	Hyp
2	$\overline{b} + a$	Hyp
3	a	Résolvant de 1, 2
4	$\overline{a} + c$	Hyp
5	c	Résolvant de 3, 4
6	$\overline{a} + \overline{d}$	Hyp
7	$\overline{d}$	Résolvant de 3, 6
8	$d + \overline{c}$	Hyp
9	d	Résolvant de 5, 8
10	$\perp$	Résolvant de 7,9

— Soit l'ensemble de clauses : $\{a + b + c, \overline{a} + b, \overline{b} + c, \overline{c} + a, \overline{a} + \overline{b} + \overline{c}\}$. Nous utilisons la même tactique pour obtenir une preuve.

1	$a+b+c$	Hyp
2	$\overline{a}+b$	Hyp
3	$b+c$	Résolvant de 1,2
4	$\overline{c}+a$	Hyp
5	$b+\overline{c}$	Résolvant de 2, 4
6	b	Résolvant de 3, 5
7	$\overline{b}+c$	Hyp
8	c	Résolvant de 6, 7
9	a	Résolvant de 4, 8
10	$\overline{a}+\overline{b}+\overline{c}$	Hyp
11	$\overline{b}+\overline{c}$	9, 10
12	$\overline{c}$	Résolvant de 6, 11
13	$\perp$	Résolvant de 8, 12

□

Exercice 4 (DPLL (20 points))

Appliquer l'algorithme `DPLL` *sur chacun des ensembles de clauses suivants. Vous donnerez la trace de l'algorithme sous forme arborescente en étiquetant clairement chaque étape par la règle utilisée et les assignations qui en découlent. Vous interpréterez clairement les résultats obtenus lorsque l'algorithme se termine et vous donnerez un modèle le cas échéant.*

1. $\{a+\overline{b}+c,\ \overline{a}+b,\ \overline{a}+\overline{c},\ a+\overline{c}+\overline{d},\ \overline{a}+c+\overline{d},\ a,\ \overline{c},\ \overline{b}+d\}$ *(5 points)*
2. $\{a+b+c+\overline{d}+\overline{e}+\overline{f},\ \overline{b}+c,\ a+c+d+f,\ \overline{a}+\overline{d}+\overline{e},\ \overline{b}+\overline{c}+\overline{f},\ \overline{b}+\overline{d}+e,\ b+\overline{c},\ b+\overline{c}+\overline{d},\ b+c+d,\ b+\overline{d},\ b+e,\ \overline{f}\}$ *(15 points)*

Réponse:

1. `a+-b+c, -a+b, -a+-c, a+-c+-d, -a+c+-d, a, -c, -b+d`

   `RE`

   `-a+b, -a+c+-d, a, -c, -b+d`

   `RU a=1, c=0`

   `b, -d, -b+d`

   `RU b=1, d=0`

   `Faux`

2. `a+b+c+-d+-e+-f, -b+c, a+c+d+f, -a+-d+-e, -b+-c+-f, -b+-d+e,`
   `b+-c, b+-c+-d, b+c+d, b+-d, b+e, -f`

   `RE`

   `-b+c, a+c+d+f, -a+-d+-e, -b+-d+e, b+-c, b+c+d, b+-d, b+e, -f`

   `RU f=0`

   `-b+c, a+c+d, -a+-d+-e, -b+-d+e, b+-c, b+c+d, b+-d, b+e`

   `Branchement sur b`

   `1) b=0`

   `a+c+d, -a+-d+-e, -c, c+d, -d, e`

```
RE

-c, c+d, -d, e

ELI e=1

-c, c+d, -d

RU c=0, d=0

FAUX

2) b=1

c, a+c+d, -a+-d+-e, -d+e

RE

c, -a+-d+-e, -d+e

ELI c=1

-a+-d+-e, -d+e

ELI a=0

-d+e

ELI d=0

Vrai
```

D'où, le modèle $a = 0, b = 1, c = 1, d = 0, f = 0$.

□

Exercice 5 (Déduction naturelle (25 points))
Prouver, **par déduction naturelle**, *les formules suivantes. Vous présenterez vos preuves sous forme de tableaux.*

1. $(C \Rightarrow \neg B) \wedge (A \Rightarrow B) \Rightarrow (C \Rightarrow \neg A)$ *(10 points)*
2. $A \wedge (\neg A \vee B) \Rightarrow A \wedge B$ *(10 points)*
3. $(\neg A \Rightarrow \neg B) \Rightarrow (B \Rightarrow A)$ *(5 points)*

Réponse:

1. $(C \Rightarrow \neg B) \wedge (A \Rightarrow B) \Rightarrow (C \Rightarrow \neg A)$

```
assume (c => -b) & (a => b).
  assume c.
    assume a.
      c => -b.
      a => b.
      -b.
      b.
      F.
    therefore -a.
  therefore c => -a.
therefore (c => -b) & (a => b) => c => -a.
```

2. $A \wedge (\neg A \vee B) \Rightarrow A \wedge B$

```
assume a & (-a + b).
  a.
  -a + b.
  assume -a.
    F.
    a & b.
  therefore -a => a & b.
  assume b.
    a & b.
  therefore b => a & b.
  a & b.
therefore a & (-a + b) => a & b.
```

3. $(\neg A \Rightarrow \neg B) \Rightarrow (B \Rightarrow A)$

```
assume -a => -b.
  assume b.
    assume -a.
      -b.
      F.
    therefore --a.
    a.
  therefore b => a.
therefore (-a => -b) => b => a.
```

□

Exercice 6 (Démonstration par récurrence (15 points))

Dans tout cet exercice on note $\oplus$ l'opérateur « ou exclusif », dont nous rappelons la table de vérité :

x	y	$x \oplus y$
0	*0*	*0*
0	*1*	*1*
1	*0*	*1*
1	*1*	*0*

— *Démontrer que le connecteur $\oplus$ est associatif, autrement dit que $(x \oplus y) \oplus z \equiv x \oplus (y \oplus z)$.*

— *Soient maintenant n variables* différentes $x_1 \dots x_n$.
Démontrer par récurrence que pour toute assignation v, on a $[x_1 \oplus x_2 \dots \oplus x_n]_v = 1$ si et seulement si le nombre de ces variables valant 1 dans v est impair.

Réponse:

— Par les tables de vérité, il apparaît clairement que les modèles de ces deux formules sont les mêmes : les assignations dans lesquelles une seule ou les trois variables valent 1.

— **Cas de base :** Si $n = 1$, on a une seule variable et la propriété est trivialement vérifiée. On peut aussi commencer à $n = 2$ sans difficulté.

Hypothèse de récurrence : Supposons que pour $n \geq 1$, on a $[x_1 \oplus \dots \oplus x_n]_v = 1$ si et seulement si le nombre de $[x_i]_v$ (i allant de 1 à n) valant 1 est impair.

Pas de récurrence Considérons $n+1$ variables différentes : $x_1 \dots x_{n+1}$.

On a 4 cas a étudier :

1. $[x_1 \oplus \dots \oplus x_n]_v = 0$ et $[x_{n+1}]_v = 0$. Dans ce cas, $[x_1 \oplus \dots \oplus x_{n+1}]_v = 0 \oplus 0 = 0$. Or, par hypothèse de récurrence, le nombre de $[x_i]_v$ (i allant de 1 à n) valant 1 est pair, ce qui reste vrai pour i de 1 à $n+1$ puisque $[x_{n+1}]_v = 0$.
2. $[x_1 \oplus \dots \oplus x_n]_v = 0$ et $[x_{n+1}]_v = 1$. Dans ce cas, $[x_1 \oplus \dots \oplus x_{n+1}]_v = 0 \oplus 1 = 1$. Or, par hypothèse de récurrence, le nombre de $[x_i]_v$ (i allant de 1 à n) valant 1 est pair, et donc il devient impair pour i de 1 à $n+1$.

3. $[x_1 \oplus \ldots \oplus x_n]_v = 1$ et $[x_{n+1}]_v = 0$. Dans ce cas, $[x_1 \oplus \ldots \oplus x_{n+1}]_v = 1 \oplus 0 = 1$. Or, par hypothèse de récurrence, le nombre de $[x_i]_v$ (i allant de 1 à n) valant 1 est impair, ce qui reste vrai pour i de 1 à $n+1$.
4. $[x_1 \oplus \ldots \oplus x_n]_v = 1$ et $[x_{n+1}]_v = 1$. Dans ce cas, $[x_1 \oplus \ldots \oplus x_{n+1}]_v = 1 \oplus 1 = 0$. Or, par hypothèse de récurrence, le nombre de $[x_i]_v$ (i allant de 1 à n) valant 1 est impair, et donc il devient pair pour i de 1 à $n+1$.

$\square$

Examen INF452

Stéphane Devismes Benjamin Wack

mai 2021
2 pages
Total : 120 points
Durée : 2h00

Documents autorisés : une feuille recto verso de notes manuscrites format A4.
Le barème est *indicatif*, les points correspondent au nombre de minutes nécessaires pour réaliser les exercices. L'épreuve sera notée sur 120 points.
Le résultat d'une question peut être admis pour s'en servir dans la suite de l'énoncé. Les exercices peuvent être traités dans l'ordre de votre choix à condition de les numéroter clairement.

Exercice 1 (Expansions (20 points))**.**

1. (10 points) *À l'aide de la méthode des expansions, déterminer* ***un contre-modèle*** *de la formule* $A = \forall x \forall y (R(x,y) \wedge R(y,x) \Rightarrow R(x,x))$.
2. (10 points) *À l'aide de la méthode des expansions, déterminer* ***un modèle*** *de la formule* $B = \forall x \exists y (P(x,y) \wedge x \neq y)$.

□

Exercice 2 (Formalisation (20 points, d'après un exercice de Christine Paulin))**.**

On cherche à modéliser un plan de table, c'est-à-dire comment les personnes sont installées pour le repas. On utilisera les symboles suivants :

— $Voisin(x,y)$ *signifie que* x *est assis à côté de* y *;*
— $Ami(x,y)$ *signifie que* x *et* y *sont amis ;*
— $Femme(x)$ *signifie que* x *est une femme.*

1. (15 points) *À l'aide de ces symboles, exprimez par des formules du premier ordre les énoncés suivants :*
   (a) *Les personnes qui sont assises côte-à-côte sont toutes amies.*
   (b) *Deux hommes ne sont jamais assis côte-à-côte.*
   (c) *Chaque femme est assise à côté d'au moins un homme.*
   (d) *Personne n'est assis tout seul.*
   (e) *On ne peut pas être assis à côté de soi-même.*
   (f) *Personne n'est assis auprès de plus de deux personnes.*
2. (5 points) *Soit le domaine* $D = \{0,1,2,3,4\}$. *On suppose que* 3 *et* 4 *sont les seules femmes. Tout le monde est ami avec tout le monde, sauf* 0 *et* 3 *qui se détestent.*

   Proposez une interprétation de la relation $Voisin$ *qui donne un modèle de l'ensemble des formules précédentes.*

□

Exercice 3 (Unification (15 points)).
Pour chacune des équations suivantes, déterminez si elle admet une solution, et si oui déterminez un unificateur le plus général. Vous utiliserez pour cela l'algorithme du cours dont vous détaillerez les étapes en donnant à chaque fois le nom des règles utilisées.

1. (5 points) $h(a, g(x, b), f(y)) = h(x, g(a, y), f(y))$
2. (5 points) $h(x, y, g(x, b)) = h(y, z, g(f(z), b))$
3. (5 points) $h(x, y, g(x, y)) = h(z, z, g(g(a, b), f(a)))$

□

Exercice 4 (Skolémisation (15 points)).
Soit A la formule suivante : $\forall x(Q(x) \Rightarrow R(x)) \wedge \exists x(P(x) \wedge Q(x)) \Rightarrow \exists x(P(x) \wedge R(x))$
Montrez que A est valide en montrant que $\neg A$ est contradictoire par la méthode de Herbrand, en suivant les étapes suivantes :

1. (10 points) *Skolémisez $\neg A$ puis mettez-la en forme clausale.*
2. (5 points) *Proposez des instances contradictoires des clauses obtenues et montrez la contradiction par résolution propositionnelle.*

□

Exercice 5 (Résolution (15 points)).
Considérons l'ensemble suivant de clauses du premier ordre :

$$\Gamma \;=\; \{R(x,y) + R(f(x),x), \quad R(x,y) + R(x,f(x)), \quad \overline{R(x,y)} + \overline{R(y,z)} + R(x,z), \quad \overline{R(y,y)}\}$$

Nous voulons montrer que cet ensemble de clauses est insatisfaisable par instanciation et par résolution.

1. (7 points) *Trouvez des instances contradictoires de ces clauses et montrez par résolution propositionnelle que ces instances sont contradictoires.*
2. (8 points) *Donnez une preuve directe du caractère contradictoire de Γ par factorisation, copie et résolution binaire. Il est possible que toutes les règles ne soient pas utilisées.*

□

Exercice 6 (Déduction naturelle (20 points)).
Démontrez les formules suivantes en déduction naturelle au premier ordre. Comme vu en cours et en TD, on numérotera les lignes de la preuve, et on indiquera pour chaque ligne la formule démontrée, le contexte, la règle appliquée et les prémisses utilisées.

1. (10 points) $\forall x \forall y \forall z(x = y \wedge \neg(y = z) \Rightarrow \neg(x = z))$
2. (10 points) $\exists x P(x) \wedge \forall x \forall y(P(x) \wedge P(y) \Rightarrow x = y) \Rightarrow \exists x(P(x) \wedge \forall y(P(y) \Rightarrow x = y))$

□

Exercice 7 (Connecteur complet (15 points, d'après un exercice du poly)).
Soit $\mid$ l'opération suivante : $x \mid y = 1$ si et seulement si $x = 0$ et $y = 0$. Cette opération est aussi appelée ni *car $x \mid y$ est vrai si et seulement si ni x, ni y ne sont vrais.*
L'objectif de cet exercice est de démontrer que ce connecteur est complet.

1. (2 points) *Donner une forme normale conjonctive et une forme normale disjonctive de la formule $x \mid y$.*
2. (2 points) *Ce connecteur est-il commutatif ($x \mid y \equiv y \mid x$) ?*
   Est-il est associatif ($(x \mid y) \mid z \equiv x \mid (y \mid z)$) ?
3. (1 point) *Donner la table de vérité de la formule $x \mid x$.*
4. (10 points) *Donner des formules équivalentes à chacune des formules : $\top$, $\bot$, $\neg x$, $x \wedge y$, $x \vee y$ et $x \Rightarrow y$ en n'utilisant que le connecteur $\mid$ et possiblement les variables x et y. On pourra justifier le résultat soit par des tables de vérité, soit en utilisant des équivalences avec les formules propositionnelles usuelles.*

□

Corrigés

Exercice 1, page 1

1. Contre-modèle par expansion :
   $D = \{0\}$: $R(0,0).R(0,0) \Rightarrow R(0,0)$ valide
   $D = \{0,1\}$: $(R(0,0).R(0,0) \Rightarrow R(0,0)).(R(0,1).R(1,0) \Rightarrow R(0,0)).(R(1,0).R(0,1) \Rightarrow R(1,1)).(R(1,1).R(1,1) \Rightarrow R(1,1))$
   Tout R_I contenant les deux couples $(0,1)$ et $(1,0)$, et faux pour au moins un couple (i,i) convient.
2. Modèle par expansion :
   $D = \{0\}$: $P(0,0).(0 \neq 0)$ est insatisfaisable.
   $D = \{0,1\}$: $(P(0,0).(0 \neq 0) + P(0,1).(0 \neq 1)).(P(1,0).(1 \neq 0) + P(1,1).(1 \neq 1)) \equiv P(0,1).P(1,0)$
   Tout P_I contenant $(0,1)$ et $(1,0)$ convient.

Exercice 2, page 1

1. Note : d'autres solutions sont admissibles, notamment selon la façon dont la symétrie des relations $Voisin$ et Ami est prise en compte.
   (a) Les personnes qui sont assises côte-à-côte sont toutes amies. $\forall x \forall y (Voisin(x,y) \Rightarrow Ami(x,y))$
   (b) Deux hommes ne sont jamais assis côte-à-côte. $\forall x \forall y (Voisin(x,y) \Rightarrow Femme(x) \vee Femme(y))$
   (c) Chaque femme est assise à côté d'au moins un homme. $\forall x (Femme(x) \Rightarrow \exists y (Voisin(x,y) \wedge \neg Femme(y)))$
   (d) Personne n'est assis tout seul. $\forall x \exists y Voisin(x,y)$
   (e) On ne peut pas être assis à côté de soi-même. $\forall x \forall y (Voisin(x,y) \Rightarrow \neg x = y)$
   (f) Personne n'est assis auprès de plus de deux personnes. $\forall x \forall y_1 \forall y_2 \forall y_3 (Voisin(x,y_1) \wedge Voisin(x,y_2) \wedge Voisin(x,y_3) \wedge \neg y_1 = y_2 \Rightarrow y_3 = y_1 \vee y_3 = y_2)$
2. On a 2 femmes et 3 hommes, comme deux hommes ne peuvent être côte-à-côte, on alterne homme-femme-homme-femme-homme. Comme 0 et 3 se détestent, la femme 3 doit être entre 1 et 2. On a donc $Voisin_I = \{(0,4),(4,1),(1,3),(3,2),(2,3),(3,1),(1,4),(4,0)\}$.
   Si on accepte 2 tables séparées, d'autres solutions du type homme-femme-homme et femme-homme sont possibles.

Exercice 3, page 2

1. $h(a, g(x,b), f(y)) = h(x, g(a,y), f(y))$
   Décomposition : $a = x$, $g(x,b) = g(a,y)$, $f(y) = f(y)$
   Identité : $a = x$, $g(x,b) = g(a,y)$
   Décomposition : $a = x$, $x = a$, $b = y$
   Élimination de x : $a = a$, $x := a$, $b = y$
   Identité : $x := a$, $b = y$
   Orientation : $x := a$, $y = b$
   Élimination de y : $x := a$, $y := b$
   $< x := a, y := b >$ est une solution la plus générale de l'équation.
2. $h(x, y, g(x,b)) = h(y, z, g(f(z), b))$
   Décomposition : $x = y$, $y = z$, $g(x,b) = g(f(z), b)$
   Élimination de x : $x := y$, $y = z$, $g(y,b) = g(f(z), b)$
   Élimination de y : $x := z$, $y := z$, $g(z,b) = g(f(z), b)$
   Décomposition : $x := z$, $y := z$, $z = f(z)$, $b = b$
   Échec de l'élimination de z pas de solution

3. $h(x, y, g(x, y)) = h(z, z, g(g(a, b), f(a)))$
   Décomposition : $x = z$, $y = z$, $g(x, y) = g(g(a, b), f(a))$
   Élimination de x : $x := z$, $y = z$, $g(z, y) = g(g(a, b), f(a))$
   Élimination de y : $x := z$, $y := z$, $g(z, z) = g(g(a, b), f(a))$
   Décomposition : $x := z$, $y := z$, $z = g(a, b)$, $z = f(a)$
   Élimination de z : $x := g(a, b)$, $y := g(a, b)$, $z := g(a, b)$, $g(a, b) = f(a)$
   Échec de la décomposition : pas de solution

Exercice 4, page 2

1. Forme normale : $\forall x(\neg Q(x) \vee R(x)) \wedge \exists x(P(x) \wedge Q(x)) \wedge \forall x(\neg P(x) \vee \neg R(x))$
2. Forme propre : $\forall x(\neg Q(x) \vee R(x)) \wedge \exists y(P(y) \wedge Q(y)) \wedge \forall z(\neg P(z) \vee \neg R(z))$
3. Elimination des $\exists$: $\forall x(\neg Q(x) \vee R(x)) \wedge P(a) \wedge Q(a) \wedge \forall z(\neg P(z) \vee \neg R(z))$
4. Enlèvement des $\forall$: $(\neg Q(x) \vee R(x)) \wedge P(a) \wedge Q(a) \wedge (\neg P(z) \vee \neg R(z))$
5. Forme clausale : $\{\neg Q(x) \vee R(x), P(a), Q(a), \neg P(z) \vee \neg R(z)\}$
6. Instances contradictoires : $\neg Q(a) \vee R(a), P(a), Q(a), \neg P(a) \vee \neg R(a)$

1 :	$\neg Q(a) \vee R(a)$	Hyp
2 :	$Q(a)$	Hyp
3 :	$R(a)$	Res 1,2
4 :	$\neg P(a) \vee \neg R(a)$	Hyp
5 :	$\neg P(a)$	Res 3,4
6 :	$P(a)$	Hyp
7 :	$\perp$	Res 5,6

Exercice 5, page 2

Instanciation : Nous obtenons les instances contradictoires suivantes :

$$\{R(a,a) + R(f(a),a), \quad R(a,a) + R(a,f(a)), \quad \overline{R(a,f(a))} + \overline{R(f(a),a)} + R(a,a), \quad \overline{R(a,a)}\}$$

Par résolution propositionnelle nous avons la preuve :

1 :	$R(a,a) + R(f(a),a)$	Hypothèse
2 :	$\overline{R(a,a)}$	Hypothèse
3 :	$R(f(a),a)$	Révolvant 1,2
4 :	$R(a,a) + R(a,f(a))$	Hypothèse
5 :	$R(a,f(a))$	Résolvant 2,4
6 :	$\overline{R(a,f(a))} + \overline{R(f(a),a)} + R(a,a)$	Hypothèse
7 :	$\overline{R(a,f(a))} + R(a,a)$	Résolvant 3,6
8 :	$R(a,a)$	Résolvant 5,7
9 :	$\perp$	Résolvant 8,2

Résolution binaire :

1 :	$R(x,y) + R(f(x),x)$	Hypothèse	
2 :	$\overline{R(y,y)}$	Hypothèse	
3 :	$\overline{R(t,t)}$	Copie 2	$< x := t >$
4 :	$R(f(t),t)$	Révolvant 1,3	$< x := t, y := t >$
5 :	$R(x,y) + R(x,f(x))$	Hypothèse	
6 :	$R(t,f(t))$	Résolvant 3,5	$< x := t, y := t >$
7 :	$\overline{R(x,y)} + \overline{R(y,z)} + R(x,z)$	Hypothèse	
8 :	$\overline{R(x,f(t))} + R(x,t)$	Résolvant 4,7	$< y := f(t), z := t >$
9 :	$R(u,f(u))$	Copie 6	$< t := u >$
10 :	$R(u,u)$	Résolvant 8,9	$< x := u, t := u >$
11 :	$\perp$	Résolvant 10,3	$< t := u >$

Exercice 6, page 2

1. $\forall x \forall y \forall z (x = y \wedge \neg(y = z) \Rightarrow \neg(x = z))$

contexte	numéro	preuve	règle
1	1	`Supposons` $x = y \wedge \neg(y = z)$	
1,2	2	`Supposons` $x = z$	
1,2	3	$x = y$	$\wedge E1$ 1
1,2	4	$y = z$	congruence 2, 3
1,2	5	$\neg(y = z)$	$\wedge E2$ 1
1,2	6	$\perp$	$\Rightarrow E$ 4, 5
1	7	`Donc` $\neg(x = z)$	$\Rightarrow I$ 2, 6
	8	`Donc` $x = y \wedge \neg(y = z) \Rightarrow \neg(x = z)$	$\Rightarrow I$ 1, 7
	9	$\forall z(x = y \wedge \neg(y = z) \Rightarrow \neg(x = z))$	$\forall I$ 8, z
	10	$\forall y \forall z(x = y \wedge \neg(y = z) \Rightarrow \neg(x = z))$	$\forall I$ 9, y
	11	$\forall x \forall y \forall z(x = y \wedge \neg(y = z) \Rightarrow \neg(x = z))$	$\forall I$ 10, x

2. $\exists x P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y) \Rightarrow \exists x (P(x) \wedge \forall y (P(y) \Rightarrow x = y))$

contexte	numéro	preuve	règle
1	1	`Supposons` $\exists x P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y)$	
1	2	$\exists x P(x)$	$\wedge E1$ 1
1,3	3	`Supposons` $P(x)$	
1,3,4	4	`Supposons` $P(y)$	
1,3,4	5	$\forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y)$	$\wedge E2$ 1
1,3,4	6	$\forall y (P(x) \wedge P(y) \Rightarrow x = y)$	$\forall E$ 5, x
1,3,4	7	$P(x) \wedge P(y) \Rightarrow x = y$	$\forall E$ 6, y
1,3,4	8	$P(x) \wedge P(y)$	$\wedge I$ 3, 4
1,3,4	9	$x = y$	$\Rightarrow E$ 8, 7
1,3	10	`Donc` $P(y) \Rightarrow x = y$	$\Rightarrow I$ 4, 9
1,3	11	$\forall y (P(y) \Rightarrow x = y)$	$\forall I$ 10, y
1,3	12	$P(x) \wedge \forall y (P(y) \Rightarrow x = y)$	$\wedge E$ 3, 11
1,3	13	$\exists x (P(x) \wedge \forall y (P(y) \Rightarrow x = y))$	$\exists I$ 12, x
1	14	`Donc` $P(x) \Rightarrow \exists x (P(x) \wedge \forall y (P(y) \Rightarrow x = y))$	$\Rightarrow I$ 3, 13
1	15	$\exists x (P(x) \wedge \forall y (P(y) \Rightarrow x = y))$	$\exists E$ 2, 14
	16	`Donc` $\exists x P(x) \wedge \forall x \forall y (P(x) \wedge P(y) \Rightarrow x = y) \Rightarrow \exists x (P(x) \wedge \forall y (P(y) \Rightarrow x = y))$	$\Rightarrow I$ 1,15

Exercice 7, page 2

1. $x \mid y = \bar{x}.\bar{y}$
2. Elle est commutative : $x \mid y \equiv \bar{x}.\bar{y} \equiv \bar{y}.\bar{x} \equiv y \mid x$
   Par contre $(1 \mid 0) \mid 0 = 0 \mid 0 = 1 \neq 0 = 1 \mid 1 = 1 \mid (0 \mid 0)$.
3. On vérifie que $x \mid x \equiv \bar{x}$.
4. $$\begin{aligned}
\top &\equiv \big(x \mid (x \mid x)\big) \mid \big(x \mid (x \mid x)\big) \\
\perp &\equiv x \mid (x \mid x) \\
\neg x &\equiv x \mid x \\
x \wedge y &\equiv (x \mid x) \mid (y \mid y) \\
x \vee y &\equiv (x \mid y) \mid (x \mid y) \\
x \Rightarrow y &\equiv \big((x \mid x) \mid y\big) \mid \big((x \mid x) \mid y\big)
\end{aligned}$$